

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to form the left and right sides of the 'A', while 'Y' characters form the central vertical bar and the top horizontal crossbar. The overall shape is a wide, symmetrical 'A' with a slightly irregular, hand-drawn appearance.

[illegible]

(1)	38	HISTORY	; DETAILED
(2)	113	DECLARATIONS	
(3)	153	EXPREG - EXPAND PROGRAM/CONTROL REGION	
(5)	372	CREPAGINI - INITIALIZE FOR CREATING DEMAND ZERO PAGES	
(6)	407	CRETVA - CREATE PAGES AT SPECIFIED VIRTUAL ADDRES	
(8)	585	CNTREG - CONTRACT PROGRAM/CONTROL REGION	
(10)	663	INADRINI, RETADRINI - INIT SCRATCH AREA AND RANGES	
(12)	784	DELTVA - DELETE VIRTUAL ADDRESS SPACE PAGES	
(14)	894	CREDEL - CREATE/DELETE COMMON LOOP	
(15)	1009	CREPAG - CREATE A SINGLE PAGE	
(16)	1185	MMG\$MOVPTLOCK - Move WSLE for PT into locked portion of WS	
(17)	1293	EXPANDCHK - EXPAND REGION CHECKING FOR SPACE AVAILABLE	
(18)	1371	UPDATPTCNT - UPDATE (MAX) PAGE TABLE COUNT	
(19)	1408	DELPAG - DELETE A SINGLE PAGE	
(21)	1988	CONTRACT - ADJUST THE LENGTH OF THE SPECIFIED REGION	
(22)	2075	PAGETYPE - CALCULATE PAGE TYPE	


```
0000 1 .TITLE SYSCREDEL - SYSTEM SERVICE CREATE & DELETE PAGE
0000 2 .IDENT 'V04-000'
0000 3 :
0000 4 :*****
0000 5 :
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :
0000 27 :++
0000 28 : FACILITY: EXECUTIVE, MEMORY MANAGEMENT SERVICES
0000 29 :
0000 30 : ABSTRACT: SYSCREPAGE IMPLEMENTS THE SERVICES WHICH CREATE NEW
0000 31 : DEMAND ZERO PAGES IN THE VIRTUAL ADDRESS SPACE OF THE CALLER.
0000 32 :
0000 33 : ENVIRONMENT: THESE SERVICES RUN IN KERNEL MODE WITH THE MEMORY
0000 34 : MANAGEMENT DATA BASE LOCKED WHEN NECESSARY.
0000 35 :
0000 36 :--
0000 37 :
0000 38 : .SBTTL HISTORY ; DETAILED
0000 39 :
0000 40 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 9-SEP-76
0000 41 :
0000 42 : MODIFIED BY:
0000 43 :
0000 44 : V03-019 WMC0014 Wayne Cardoza 28-Mar-1984
0000 45 : Only window pages above MAXPFN cause process to be locked
0000 46 : on primary.
0000 47 :
0000 48 : V03-018 WMC0013 Wayne Cardoza 25-Feb-1983
0000 49 : Handle deletion of resident global pages.
0000 50 :
0000 51 : V03-017 WMC0012 Wayne Cardoza 16-Jan-1983
0000 52 : CRECOM_DONE path must check for SS$_ABORT.
0000 53 :
0000 54 : V03-016 WMC0011 Wayne Cardoza 28-Sep-1983
0000 55 : Move code to FAST_CREATE so CRMPSC can use it.
0000 56 :
0000 57 : V03-015 WMC0010 Wayne Cardoza 26-Sep-1983
```


0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :
0000 69 :
0000 70 :
0000 71 :
0000 72 :
0000 73 :
0000 74 :
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 :
0000 80 :
0000 81 :
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 :
0000 87 :
0000 88 :
0000 89 :
0000 90 :
0000 91 :
0000 92 :
0000 93 :
0000 94 :
0000 95 :
0000 96 :
0000 97 :
0000 98 :
0000 99 :
0000 100 :
0000 101 :
0000 102 :
0000 103 :
0000 104 :
0000 105 :
0000 106 :
0000 107 :
0000 108 :
0000 109 :
0000 110 :
0000 111 :

Make sure 1 page EXPREG of P1 space returns addresses in correct order.

V03-014 WMC0009 Wayne Cardoza 28-Aug-1983
Don't wait for I/O completion at IPL 2.
Major changes to creation of demand zero pages to improve performance.

V03-013 WMC0008 Wayne Cardoza 22-Aug-1983
MOVPTLOCK - don't remove pages from permanent working set.

V03-012 WMC0007 Wayne Cardoza 03-Aug-1983
Misc performance improvements.
New entry point for EXPREG with arbitrary protection.

V03-011 WMC0006 Wayne Cardoza 8-Jun-1983
Properly delete demand-zero global page file pages.

V03-010 TCM0001 Trudy C. Matthews 4-Apr-1983
Change references to working set fields in PHD so that they are used as unsigned words.

V03-009 WMC0005 Wayne Cardoza 18-Mar-1983
Fix status returned by delete.

V03-008 WMC0004 Wayne Cardoza 02-Mar-1983
MMG\$CRECOMx entry points go away.
MMG\$RETADRINI, MMG\$INADRINI, MMG\$RETRANGE will always return status and with an RSB.

V03-007 WMC0003 Wayne Cardoza 02-Feb-1983
Make MMG\$CREDEL and MMG\$RETRANGE global entry points.

V03-006 WMC0002 Wayne Cardoza 20-Jan-1983
Don't write back a process section page if the checkpoint bit is turned on in the backing store word.

V03-005 KDM0027 Kathleen D. Morse 10-Nov-1982
Invalidate the translation buffer upon deletion of a shared memory global section page. Do not clear window bit in PTE until sure that deletion can proceed in case I/O wait causes re-execution of \$DELPAG.

V03-004 MLJ0101 Martin L. Jack 13-Nov-1982
Fix broken BSBW.

V03-003 WMC0001 Wayne Cardoza 29-Sep-1982
Give up if ADJWSL doesn't change extradyn.

V03-002 KDM46395 Kathleen D. Morse 25-Jun-1982
Place the WSLEs for page table pages that contain PFNMAP and MA780 global pages, into the locked portion of the working set.

```
0000 113      .SBTTL  DECLARATIONS
0000 114      :
0000 115      : INCLUDE FILES:
0000 116      :
0000 117      $IPLDEF      ;PROCESSOR PRIORITY LEVELS
0000 118      $IRPDEF      ;I/O REQUEST PACKET DEFINITIONS
0000 119      $JIBDEF      ;JOB INFORMATION BLOCK DEFINITIONS
0000 120      $MMGDEF      ; Offsets from FP into scratch area
0000 121      $OPDEF       ; Define opcode equivalent values
0000 122      $PCBDEF      ;PROCESS CONTROL BLOCK DEFINITIONS
0000 123      $PFNDEF      ;PAGE FRAME NUMBER DATA BASE
0000 124      $PHDDEF      ;PROCESS HEADER DEFINITIONS
0000 125      $PRDEF       ;PROCESSOR REGISTER DEFINITIONS
0000 126      $PSLDEF      ;PSL
0000 127      $PTEDEF      ;PAGE TABLE ENTRY DEFINITIONS
0000 128      $RSNDEF      ;RESOURCE WAIT NUMBER DEFINITIONS
0000 129      $SECDEF      ;SECTION TABLE DEFINITIONS
0000 130      $SSDEF       ;SYSTEM STATUS CODE DEFINITIONS
0000 131      $VADEF       ;VIRTUAL ADDRESS VIELDS
0000 132      $WSLDEF      ;WORKING SET LIST ENTRY DEFINITIONS
0000 133      :
0000 134      : EXTERNAL SYMBOLS:
0000 135      :
0000 136      :
0000 137      : MACROS:
0000 138      :
0000 139      :
0000 140      : EQUATED SYMBOLS:
0000 141      :
0000 142      : OFFSET FROM AP
0000 143      :
00000004 0000 144      INADR      = 4      ;OFFSET TO INPUT RANGE
00000004 0000 145      PAGCNT     = 4      ;OFFSET TO PAGE COUNT
00000008 0000 146      RETADR     = 8      ;OFFSET TO RETURN RANGE
0000000C 0000 147      ACMODE     = 12     ;ACCESS MODE
00000010 0000 148      REGION     = 16     ;OFFSET TO REGION
0000 149      :
0000 150      : OWN STORAGE:
0000 151      :
```



```
0000 153 .SBTTL EXPREG - EXPAND PROGRAM/CONTROL REGION
0000 154
0000 155 :++
0000 156 : FUNCTIONAL DESCRIPTION:
0000 157
0000 158 : THE EXPAND REGION SYSTEM SERVICE EXPANDS THE PROGRAM REGION (REGION = 0)
0000 159 : OR THE CONTROL REGION (REGION = 1) OF THE PROCESS SPACE BY THE
0000 160 : NUMBER OF PAGES SPECIFIED IN THE PAGCNT PARAMETER. THE NEW PAGES
0000 161 : ARE DEMAND ZERO PAGES OWNED BY THE MODE OF THE CALLER WITH
0000 162 : READ/WRITE PROTECTION FOR THE CALLING MODE AND ALL MORE PRIVILEGED MODES.
0000 163
0000 164 : CALLING SEQUENCE:
0000 165
0000 166 : CALLG ARGLIST,@#SYS$EXPREG
0000 167 : CALLG ARGLIST,MMG$EXPREG ;PRIVILEGED, NON VECTORED ENTRY POINT
0000 168 : ;ACCEPTS DESIRED PAGE PROTECTION IN ACMODE+1
0000 169 : ;PRESERVES IPL IF LEQ ASTDEL
0000 170
0000 171 : INPUT PARAMETERS:
0000 172
0000 173 : PAGCNT(AP) = NUMBER OF PAGES TO ADD TO THE REGION
0000 174 : RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0000 175 : THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0000 176 : OF THE PAGES JUST CREATED
0000 177 : ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH THE CALLERS MODE
0000 178 : USED AS THE OWNER OF THE NEW PAGE(S)
0000 179 : ACMODE+1(AP) = THE PROTECTION TO USE FOR THE NEW PAGE WHEN
0000 180 : CALLED AT THE PRIVILEGED MMG$EXPREG ENTRY POINT
0000 181 : REGION(AP) = THE INDICATOR FOR WHICH REGION TO EXPAND
0000 182 : 0 = PROGRAM REGION
0000 183 : 1 = CONTROL REGION
0000 184
0000 185 : IMPLICIT INPUTS:
0000 186
0000 187 : CURRENT PCB LOCATED VIA SCH$GL_CURPCB
0000 188 : PROCESS HEADER AS SPECIFIED IN PCB$$_PHD OF THE PCB
0000 189
0000 190 : OUTPUT PARAMETERS:
0000 191
0000 192 : R0 = SYSTEM STATUS CODE
0000 193
0000 194 : IMPLICIT OUTPUTS:
0000 195
0000 196 : NONE
0000 197
0000 198 : COMPLETION CODES:
0000 199
0000 200 : SSS_NORMAL ;SUCCESSFUL COMPLETION
0000 201 : SSS_ACCVIO ;RETURN ADDRESS RANGE INACCESSIBLE
0000 202 : SSS_VASFULL ;VIRTUAL ADDRESS SPACE FULL
0000 203 : SSS_ILLPAGCNT ;ILLEGAL PAGE COUNT PARAMETER
0000 204
0000 205 : SIDE EFFECTS:
0000 206
0000 207 : NONE
0000 208
0000 209 :--
```

```
0000 211 :
0000 212 : *****
0000 213 :
0000 214 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0000 215 :
0000 216 : .PSECT Y$EXEPAGED
0000 217 :
0000 218 : *****
0000 219 :
0000 220 :
0000 221 : .ENABL LSB
0000 222 :
0000 223 MMG$EXPREG::
57 0D AC 01FC 0000 224 .WORD ^M<R2,R3,R4,R5,R6,R7,R8>
0A 11 0002 225 MOVZBL ACMODE+1(AP),R7 ;PICK UP DESIRED PAGE PROTECTION IF SPECIFIE
0006 226 BRB 8$
0008 227
50 00FC 8F 3C 0008 228 5$: MOVZWL #SS$_ILLPAGCNT,R0 ;ILLEGAL PAGE COUNT PARAM
04 000D 229 6$: RET
000E 230
01FC 000E 231 .ENTRY EXE$EXPREG,^M<R2,R3,R4,R5,R6,R7,R8>
0010 232
57 D4 0010 233 CLRL R7 ;NO PAGE PROTECTION SPECIFIED
SE 1C C2 0012 234 8$: SUBL #-MMG$C_LENGTH,SP ;RESERVE SCRATCH AREA
028E 30 0015 235 BSBW MMG$RETADRINI ;INIT RETURN RANGE AND SCRATCH AREA
F2 50 E9 0018 236 BLBC R0,6$
013E 30 001B 237 BSBW MMG$CREPAGINI ;INIT FOR CREATING DZRO PAGES
57 D5 001E 238 TSTL R7 ;PAGE PROTECTION SPECIFIED?
05 13 0020 239 BEQL 9$ ;BRANCH IF NOT
58 04 1B 57 F0 0022 240 INSV R7,#PTESV_PROT,#PTES$_PROT,R8 ;SET DESIRED PAGE PROTECTION
56 04 AC D0 0027 241 9$: MOVL PAGCNT(AP),R6 ;PAGES TO BE ALLOCATED
51 56 01 C3 002B 242 SUBL3 #1,R6,R1 ;R1=# PAGES TO BE ALLOCATED - 1
51 FFE00000 8F D3 002F 243 BITL ^C<VASM_VPG & - ;PAGE COUNT TOO LARGE?
0036 244 ^C<VASM_P1 ! VASM_SYSTEM>>a-9,R1 ;(HIGH 11 BITS NOT CLEAR?)
0036 245 BNEQ 5$ ;BRANCH FOR ERROR IF NOT CLEAR
51 51 09 78 0038 246 ASHL #9,R1,R1 ;EXPRESS AS BYTES
003C 247 :
003C 248 : ***** LOCK THE PROCESS HEADER HERE
003C 249 :
003C 250 :
54 00000000'EF D0 003F 251 SETIPL #IPL$ ASTDEL ;NO AST'S WHILE MODIFYING PHD
55 00000000'9F D0 0046 252 MOVL SCH$GC_CURPCB,R4 ;R4=PCB
0A 10 AC E8 004D 253 MOVL @#CTL$GL_PHD,R5 ;R5=PHD ADR
52 28 A5 D0 0051 254 BLBS REGION(AP),10$ ;BRANCH IF P1 SPACE
53 52 51 C1 0055 255 MOVL PHD$1_FREPOVA(R5),R2 ;STARTVA=1ST FREE P0 SPACE PAGE
08 11 0059 256 ADDL3 R1,R2,R3 ;R3=ENDVA (INCLUSIVE)
52 30 A5 D0 005B 257 10$: BRB 15$
53 52 51 C3 005F 258 MOVL PHD$1_FREP1VA(R5),R2 ;STARVA=1ST FREE P1 SPACE PAGE
0063 259 SUBL3 R1,R2,R3 ;R3=ENDVA (INCLUSIVE)
51 0200 C1 DE 0063 260 15$: MOVAL ^X200(R1),R1 ;GET FULL SIZE OF NEW REGION
0022 30 0068 261 BSBW MMG$TRY_ALL ;SEE IF FULL EXPANSION IS POSSIBLE
05 50 E9 006B 262 BLBC R0,100$ ;GO TRY IT THE HARD WAY
006C 30 006E 263
08 11 0071 264 BSBW MMG$FAST_CREATE ;USE THE FAST CREATE PATH
0073 265 BRB 105$
0073 266 :
0073 267 : GET AS FAR AS WE CAN - ONE PAGE AT A TIME
```



```
56 0414'CF DE 0073 268 :  
    02CC 30 0073 269 100$: MOVAL W^MMG$CREPAG,R6 ;MMG$CREDEL NEEDS THIS  
    50 DD 0078 270 BSBW MMG$CREDEL ;COMMON CREATE/DELETE LOOP  
    0242 30 007B 271 105$: PUSHL RO ;SAVE STATUS  
    03 50 E8 007D 272 BSBW MMG$RETRANGE ;RETURN RANGE OPERATED ON  
    6E 50 D0 0080 273 BLBS RO,110$  
    50 BA 0083 274 MOVL RO,(SP) ;THIS STATUS WILL SUPERCEDE CREDEL  
    50 0086 275 110$: POPR RO  
    0088 276 ENBINT MMG$_CALLEDIPL(FP)  
    04 008C 277 RET  
    008D 278  
    008D 279 .DSABL LSB  
    008D 280  
    008D 281 : TRY EXPANDING REGION TO FIT LAST PAGE TO BE CREATED  
    008D 282 : THEN CHECK AND ADJUST PAGEFILE QUOTA  
    008D 283  
    008D 284 : R1 -> ADDITIONAL SPACE REQUIRED  
    008D 285 : R2 -> A VA IN THE REGION  
    008D 286 : R4 -> PCB  
    008D 287 : R6 -> PAGES TO BE CREATED (MAY BE LESS THAN REQUIRED REGION EXPANSION)  
    008D 288  
    008D 289 MMG$TRY_ALL::  
    47 50 30 008D 290 BSBW EXPANDCHK ;TRY EXPANDING REGION  
    0090 291 BLBC RO,70$ ;NO LUCK  
    0093 292 :  
    08 E1 0093 293 BBC #MMG$V CHGPAGFIL,- ;BRANCH IF NOT CHARGING FOR PAGE FILE  
    3E FC AD 0095 294 MMG$_MAXACMODE(FP),60$  
    50 0080 C4 D0 0098 295 MOVL PCB$_JIB(R4),RO ;GET JIB ADDRESS  
    56 3C A0 D1 009D 296 CMPL JIB$_PGFLCNT(RO),R6 ;CHECK QUOTA  
    37 19 00A1 297 BLSS 70$ ;BRANCH IF OVER QUOTA  
    3C A0 56 C2 00A3 298 R6,JIB$_PGFLCNT(RO)  
    50 38 A0 3C A0 C3 00A7 299 JIB$_PGFLCNT(RO),JIB$_PGFLQUOTA(RO),RO ;PAGE FILE PAGES IN USE  
    00000000'GF 50 D1 00AD 300 CMPL RO,G^CTL$GL_VIRTPEAK ;MAINTAIN MAX PAGE FILE USED  
    10 1F 00B4 301 BLSSU 50$  
    00000000'GF 50 D0 00B6 302 MOVL RO,G^CTL$GL_VIRTPEAK ;SET NEW PEAK VALUE  
    00000000'GF 50 D0 00BD 303 MOVL RO,G^CTL$GL_IPAGEFL ;SET NEW PEAK VALUE  
    10 11 00C4 304 BRB 60$  
    00000000'GF 50 D1 00C6 305 50$: CMPL RO,G^CTL$GL_IPAGEFL ;MAINTAIN MAX PAGE FILE USED (IMAGE)  
    07 1F 00CD 306 BLSSU 60$  
    00000000'GF 50 D0 00CF 307 MOVL RO,G^CTL$GL_IPAGEFL ;SET NEW PEAK VALUE  
    00D6 308  
    50 01 3C 00D6 309 60$: MOVZWL #SS$_NORMAL,RO  
    05 00D9 310 RSB  
    00DA 311  
    50 D4 00DA 312 70$: CLRL RO ;CAN'T MAKE IT  
    05 00DC 313 RSB  
    00DD 314  
    00DD 315  
    00DD 316 : FAST CREATE PATH  
    00DD 317  
    00DD 318 : R2 -> STARTVA  
    00DD 319 : R3 -> ENDVA  
    00DD 320 : R5 -> P1 PHD  
    00DD 321 : R6 -> PAGE COUNT  
    00DD 322 : R8 -> NEW PTE  
    00DD 323  
    00DD 324 MMG$FAST_CREATE::
```

```

      7E 52 7D 00DD 325      MOVQ R2,-(SP)      ;SAVE THE VA'S
      53 52 D1 00E0 326      CMPL R2,R3        ;IS R2 THE LOWEST VA
      03 1F 00E3 327      BLSSU 10$          ;YES
      52 53 D0 00E5 328      MOVL R3,R2        ;NOW IT IS
      07 52 1E E1 00E8 329      :
      00EC 331      : P1 SPACE
      00EC 332      :
      00EC 333      :
      53 00D0 C5 DE 00EC 334      MOVAL PHD$L_P1BR(R5),R3 ;ADR OF POINTER TO P1PT
      05 11 00F1 335      BRB 30$
      00F3 336      :
      00F3 337      : P0 SPACE
      00F3 338      :
      53 00C8 C5 DE 00F3 339 20$: MOVAL PHD$L_POBR(R5),R3 ;ADR OF POINTER TO POPT
      00F8 340      :
      51 52 15 09 EF 00F8 341 30$: EXTZV #VASV_VPN,#VASS_VPN,R2,R1 ;VIRTUAL PAGE NUMBER
      50 56 02 00 EF 00FD 342      EXTZV #0,#2,R6,R0 ;PAGES TO LEAVE A MULTIPLE OF 4
      0A 13 0102 343      BEQL 50$
      00 B341 58 D0 0104 344 40$: MOVL R8,@(R3)[R1] ;STORE NEW PTE
      51 D6 0109 345      INCL R1 ;NEXT PAGE
      F6 50 F5 010B 346      SOBGTR R0,40$
      56 56 FE 8F 78 010E 347 50$: ASHL #-2,R6,R6
      1F 13 0113 348      BEQL 70$ ;DONE
      00 B341 58 D0 0115 349 60$: MOVL R8,@(R3)[R1] ;STORE NEW PTE
      51 D6 011A 350      INCL R1 ;NEXT PAGE
      00 B341 58 D0 011C 351      MOVL R8,@(R3)[R1] ;STORE NEW PTE
      51 D6 0121 352      INCL R1 ;NEXT PAGE
      00 B341 58 D0 0123 353      MOVL R8,@(R3)[R1] ;STORE NEW PTE
      51 D6 0128 354      INCL R1 ;NEXT PAGE
      00 B341 58 D0 012A 355      MOVL R8,@(R3)[R1] ;STORE NEW PTE
      51 D6 012F 356      INCL R1 ;NEXT PAGE
      E1 56 F5 0131 357      SOBGTR R6,60$
      0134 358      :
      51 8E 7D 0134 359 70$: MOVQ (SP)+,R1 ;GET BACK THE VA'S
      52 51 D1 0137 360      CMPL R1,R2
      12 1A 013A 361      BGTRU 90$ ;GOING BACKWARDS
      04 1F 013C 362      BLSSU 80$ ;GOING FORWARDS
      OC 52 1E E0 013E 363      BBS #VASV_P1,R2,90$ ;EQUAL - P1 IS BACKWARDS
      51 01FF 8F AA 0142 364 80$: BICW #^X1FF,R1 ;STARTVA IS START OF PAGE
      52 01FF 8F A8 0147 365      BISW #^X1FF,R2 ;ENDVA IS END OF PAGE
      0A 11 014C 366      BRB 100$
      52 01FF 8F AA 014E 367 90$: BICW #^X1FF,R2 ;STARTVA IS START OF PAGE
      51 01FF 8F A8 0153 368      BISW #^X1FF,R1 ;ENDVA IS END OF PAGE
      50 01 3C 0158 369 100$: MOVZWL #SS$_NORMAL,R0 ;INDICATE SUCCESSFUL COMPLETION
      05 015B 370      RSB
```



```
015C 372      .SBTTL CREPAGINI - INITIALIZE FOR CREATING DEMAND ZERO PAGES
015C 373
015C 374 :
015C 375 : INPUTS:
015C 376 :
015C 377 : MMG$L_MAXACMODE(FP) CONTAINS THE MAXIMIZED ACCESS MODE
015C 378 :
015C 379 : OUTPUTS:
015C 380 :
015C 381 : R6 = ADDRESS OF CREPAG SUBROUTINE TO BE CALLED
015C 382 : R8 = DEMAND ZERO PAGE TABLE ENTRY FOR SPECIFIED ACCESS MODE
015C 383 :
015C 384 :
015C 385 : *****
015C 386 :
015C 387 : ***** THE FOLLOWING CODE MAY BE PAGED *****
015C 388 :
0000015C 389 : .PSECT Y$EXEPAGED
015C 390 :
015C 391 : *****
015C 392 :
015C 393 MMG$CREPAGINI:
015C 394      MOVAL W^MMG$CREPAG,R6      ;ADR OF INITITAL DELETE SCANNER FOR CREATE
56      0414'CF DE 015C 395      MOVZBL MMG$L_MAXACMODE(FP),R8 ;ACCESS MODE OF WRITER/OWNER
58      FC AD 9A 0161 396      BBSS #MMG$V_CHGPAGFIL,MMG$L_MAXACMODE(FP),10$ ;CHARGE PAGE FILE QUOTA
00 FC AD 08 E2 0165 397 10$: MOVZBL B^MMG$AB_DZRO[R8],R8 ;PROTECTION AND OWNER FIELDS
58      74'AF48 9A 016A 398      ASHL #PTESV_OWN,R8,R8 ;SHIFT INTO POSITION
58      58 17 78 016F 399      RSB
015C 400
015C 401 MMG$AB_DZRO:
20 0174 402      .BYTE <PTESC_KW ! PTESC_KOWN>@-PTESV_OWN ;KERNEL
51 0175 403      .BYTE <PTESC_EW ! PTESC_EOWN>@-PTESV_OWN ;EXEC
82 0176 404      .BYTE <PTESC_SW ! PTESC_SOWN>@-PTESV_OWN ;SUPER
43 0177 405      .BYTE <PTESC_UW ! PTESC_UOWN>@-PTESV_OWN ;USER
```

```
0178 407 .SBTTL CRETVA - CREATE PAGES AT SPECIFIED VIRTUAL ADDRES
0178 408
0178 409 :++
0178 410 : FUNCTIONAL DESCRIPTION:
0178 411 : THE CREATE VIRTUAL ADDRESS SPACE SYSTEM SERVICE CREATES NEW DEMAND
0178 412 : ZERO PAGES IN THE SPECIFIED RANGE OF VIRTUAL ADDRESSES.
0178 413 : IF ANY PAGES ALREADY EXIST IN THE RANGE, THEY ARE DELETED
0178 414 : THE NEW PAGES ARE READ/WRITE TO THE CALLING MODE AND ALL MORE
0178 415 : PRIVILEGED MODES.
0178 416 :
0178 417 : CALLING SEQUENCE:
0178 418 :
0178 419 : CALLG  ARGLIST,@#SYSS$CRETVA
0178 420 : CALLG  ARGLIST,MMG$CRETVA ;PRIVILEGED, NON VECTORED ENTRY POINT
0178 421 : ;ACCEPTS DESIRED PAGE PROTECTION IN ACMODE+1
0178 422 : ;PRESERVES IPL IF LEQ ASTDEL
0178 423 :
0178 424 :
0178 425 : INPUT PARAMETERS:
0178 426 :
0178 427 : INADR(AP) = ADDRESS OF 2 LONG WORDS THE 1ST OF WHICH SPECIFIES
0178 428 : THE STARTING VIRTUAL ADDRESS TO CREATE, THE 2ND SPECIFIES
0178 429 : THE ENDING VIRTUAL ADDRESS TO CREATE (INCLUSIVE).
0178 430 : RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0178 431 : THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0178 432 : OF THE PAGES JUST CREATED
0178 433 : ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH CALLING MODE)
0178 434 : USED AS THE OWNER OF THE NEW PAGE(S)
0178 435 : ACMODE+1(AP) = THE PROTECTION TO USE FOR THE NEW PAGE WHEN
0178 436 : CALLED AT THE PRIVILEGED MMG$CRETVA ENTRY POINT
0178 437 :
0178 438 : IMPLICIT INPUTS:
0178 439 :
0178 440 : NONE
0178 441 :
0178 442 : OUTPUT PARAMETERS:
0178 443 :
0178 444 : RO = SYSTEM STATUS CODE
0178 445 :
0178 446 : IMPLICIT OUTPUTS:
0178 447 :
0178 448 : NONE
0178 449 :
0178 450 : COMPLETION CODES:
0178 451 :
0178 452 : SSS_NORMAL ;SUCCESSFUL COMPLETION
0178 453 : SSS_ACCVIO ;ACCESS VIOLATION
0178 454 : SSS_NOPRIV ;NO PRIVILEGE TO CREATE/DELETE PAGE
0178 455 : SSS_VASFULL ;VIRTUAL ADDRESS SPACE FULL
0178 456 :
0178 457 : SIDE EFFECTS:
0178 458 :
0178 459 : NONE
0178 460 :
0178 461 :--
```



```
0178 463 :
0178 464 : *****
0178 465 :
0178 466 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0178 467 :
0000 0178 468 .PSECT Y$EXEPAGED
0178 469 :
0178 470 : *****
0178 471 :
0178 472 :
0178 473 .ENABL LSB
0178 474 :
50 24 3C 0178 475 5$: MOVZWL #SS$ NOPRIV,R0 ;NO PRIVILEGE
F4 AD D4 017B 476 CLRL MMG$_SAVRETADR(FP) ;NULL RETURN ADR FOR NULL RANGE
04 017E 477 6$: RET
017F 478 :
017F 479 MMG$CRETVA::
017F 480 .WORD ^M<R2,R3,R4,R5,R6,R7,R8>
57 0D AC 9A 0181 481 MOVZBL ACMODE+1(AP),R7 ;PICK UP DESIRED PAGE PROTECTION IF SPECIFIE
04 11 0185 482 BRB 10$
0187 483 :
01FC 0187 484 .ENTRY EXE$CRETVA,^M<R2,R3,R4,R5,R6,R7,R8>
0189 485 :
57 D4 0189 486 CLRL R7 ;NO PAGE PROTECTION SPECIFIED
5E 1C C2 018B 487 10$: SUBL #-MMG$C LENGTH,SP ;RESERVE SCRATCH AREA
0106 30 018E 488 BSBW MMG$INADRINI ;GET INPUT RANGE TO R4, R5
0191 489 ;INIT RETURN RANGE AND SCRATCH AREA
EA 50 E9 0191 490 BLBC R0,6$
FFC5 30 0194 491 BSBW MMG$CREPAGINI ;INIT FOR CREATING DZRO PAGES
57 D5 0197 492 TSTL R7 ;PAGE PROTECTION SPECIFIED?
05 13 0199 493 BEQL 20$ ;BRANCH IF NOT
58 04 1B 57 F0 019B 494 INSV R7,#PTESV_PROT,#PTES$_PROT,R8 ;SET DESIRED PAGE PROTECTION
52 54 7D 01A0 495 20$: MOVQ R4,R2 ;R2 = START OF RANGE, R3 = END
01A3 496 :
01A3 497 : OPERATE ON PAGES STARTING WITH THE ADDRESS IN R2 ENDING WITH THE
01A3 498 : ADDRESS IN R3 INCLUSIVE.
01A3 499 :
01A3 500 :
01A3 501 : ***** LOCK THE PROCESS HEADER HERE
01A3 502 :
01A3 503 SETIPL #IPL$_ASTDEL ;NO AST'S WHILE MODIFYING PHD
01A6 504 :
CE 52 1F E0 01A6 505 BBS #31,R2,5$ ;CHECK FOR SYSTEM SPACE ADDRESS
CA 53 1F E0 01AA 506 BBS #31,R3,5$ ;CHECK FOR SYSTEM SPACE ADDRESS
2F 10 01AE 507 BSBB MMG$IN REGION ;IS REQUESTED SPACE IN EXITING SPACE
12 50 E9 01B0 508 BLBC R0,100$ ;YES - DO CREATE THE HARD WAY
01B3 509 :
54 00000000'EF D0 01B3 510 MOVL SCH$GL_CURPCB,R4
FED0 30 01BA 511 BSBW MMG$TRY_ALL ;CHECK EXPANSION AND QUOTAS
05 50 E9 01BD 512 BLBC R0,100$ ;NO LUCK - DO IT THE SLOW WAY
FF1A 30 01C0 513 BSBW MMG$FAST_CREATE ;FILL IN THE PTE'S
08 11 01C3 514 BRB 110$
01C5 515 :
01C5 516 :
01C5 517 : R2 = STARTING VA, R3 = ENDING VA
01C5 518 : R6 = SUBROUTINE TO CALL, R8 = PROTECTION
01C5 519 :
```

```
56 0414'CF DE 01C5 520 100$: MOVAL W*MMG$CREPAG,R6 ;MMG$CREDEL NEEDS THIS
    017A 30 01CA 521 BSBW MMG$CREDEL ;COMMON CREATE/DELETE LOOP
    50 DD 01CD 522 110$: PUSHL R0 ;SAVE STATUS
    00F0 30 01CF 523 BSBW MMG$RETRANGE ;RETURN RANGE OPERATED ON
    03 50 E8 01D2 524 BLBS R0,60$
    6E 50 D0 01D5 525 MOVL R0,(SP) ;THIS STATUS WILL SUPERCEDE CREDEL
    50 BA 01D8 526 60$: POPR R0
    04 01DA 527 ENBINT MMG$L_CALLEDIPL(FP) ;RESTORE IPL AT CALL
    01DE 528 RET
    01DF 529
    01DF 530 .DSABL LSB
    01DF 531 :
    01DF 532 :
    01DF 533 : CHECK TO SEE IF ADDRESS SPACE OVERMAPS EXISTING SPACE
    01DF 534 :
    01DF 535 : INPUTS
    01DF 536 : R4 -> STARTING VA
    01DF 537 : R5 -> ENDING VA
    01DF 538 : OUTPUTS
    01DF 539 : R0 -> SUCCESS OR FAILURE
    01DF 540 : R1 -> BYTES TO EXPAND REGION
    01DF 541 : R5 -> P1 PHD
    01DF 542 : R6 -> PAGE COUNT OF VA REQUEST (NOT EXPANSION SIZE)
    01DF 543 :
    01DF 544 MMG$IN_REGION::
    54 01FF 8F AA 01DF 545 BICW #*X1FF,R4 ; GET THE REQUESTED PAGE COUNT
    55 01FF 8F AA 01E4 546 BICW #*X1FF,R5
    56 55 54 C3 01E9 547 SUBL3 R4,R5,R6
    56 56 F7 8F 78 01ED 548 ASHL #-9,R6,R6 ;GET A PAGE COUNT
    03 18 01F2 549 BGEQ 30$
    56 56 CE 01F4 550 MNEGL R6,R6 ;MAKE IT A POSITIVE NUMBER
    56 56 D6 01F7 551 30$: INCL R6 ;IT WAS SHORT BY ONE
    01F9 552 :
    01F9 553 : DECIDE IF THIS IS ALL PAST CURRENT END OF REGION
    01F9 554 :
    55 51 55 D0 01F9 555 MOVL R5,R1 ;ASSUME THIS IS FURTHEST OUT
    00000000'9F D0 01FC 556 MOVL @#CTL$GL_PHD,R5 ;P1 PHD ADDRESS
    1B 51 1E E0 0203 557 BBS #VASV_P1,R1,40$
    50 28 A5 D0 0207 558 MOVL PHD$L_FREPOVA(R5),R0 ;FIRST FREE P0
    50 54 D1 020B 559 CMPL R4,R0
    35 19 020E 560 BLSS 100$ ;NOT IN FREE SPACE
    51 54 D1 0210 561 CMPL R4,R1 ;WHICH IS REALLY FURTHER OUT
    08 19 0213 562 BLSS 35$
    50 51 D1 0215 563 CMPL R1,R0 ;R1 WAS REALLY CLOSER
    2B 19 0218 564 BLSS 100$ ;NOT IN FREE SPACE
    51 54 D0 021A 565 MOVL R4,R1 ;WE WERE WRONG THE FIRST TIME
    51 50 C2 021D 566 35$: SUBL R0,R1 ;BYTES NEEDED (ONE PAGE SHORT)
    1A 11 0220 567 BRB 50$
    50 30 A5 D0 0222 568 40$: MOVL PHD$L_FREP1VA(R5),R0 ;FIRST FREE P1
    50 54 D1 0226 569 CMPL R4,R0
    1A 14 0229 570 BGTR 100$ ;NOT IN FREE SPACE
    51 54 D1 022B 571 CMPL R4,R1 ;WHICH IS REALLY FURTHER OUT
    08 14 022E 572 BGTR 45$
    50 51 D1 0230 573 CMPL R1,R0 ;R1 WAS REALLY CLOSER
    10 14 0233 574 BGTR 100$ ;NOT IN FREE SPACE
    51 51 54 D0 0235 575 MOVL R4,R1 ;WE WERE WRONG THE FIRST TIME
    51 50 51 C3 0238 576 45$: SUBL3 R1,R0,R1 ;BYTES NEEDED (ONE PAGE SHORT)
```


SYSCREDEL
V04-000

- SYSTEM SERVICE CREATE & DELETE PAGE
CRETVA - CREATE PAGES AT SPECIFIED VIRTU

E 11

16-SEP-1984 01:49:03 VAX/VMS Macro V04-00
5-SEP-1984 03:49:40 [SYS.SRC]SYSCREDEL.MAR;1

Page 12
(7)

51	0200	C1	DE	023C	577	50\$:	MOVAL	^X200(R1),R1	:ADD IN THE EXTRA PAGE
	50	01	D0	0241	578		MOVL	#1,R0	:SUCCESS
			05	0244	579		RSB		
				0245	580				
	50		D4	0245	581	100\$:	CLRL	R0	:FAILURE
			05	0247	582		RSB		
				0248	583				

```
0248 585 .SBTTL CNTREG - CONTRACT PROGRAM/CONTROL REGION
0248 586 :++
0248 587 : FUNCTIONAL DESCRIPTION:
0248 588 :
0248 589 :
0248 590 : CALLING SEQUENCE:
0248 591 :
0248 592 : CALLG  ARGLIST,@#SYSS$CNTREG
0248 593 :
0248 594 :
0248 595 : INPUT PARAMETERS:
0248 596 :
0248 597 :     PAGCNT(AP)      =      NUMBER OF PAGES TO DELETE
0248 598 :     RETADR(AP)     =      RETURN ADDRESS RANGE OF PAGES DELETED
0248 599 :     ACMODE(AP)     =      ACCESS MODE TO CHECK AGAINST PAGE OWNER
0248 600 :     REGION(AP)     =      0 FOR PROGRAM REGION
0248 601 :                     =      1 FOR CONTROL REGION
0248 602 :
0248 603 : IMPLICIT INPUTS:
0248 604 :     NONE
0248 605 :
0248 606 : OUTPUT PARAMETERS:
0248 607 :
0248 608 :     RO = SYSTEM STATUS CODE
0248 609 :
0248 610 : IMPLICIT OUTPUTS:
0248 611 :     NONE
0248 612 :
0248 613 : COMPLETION CODES:
0248 614 :
0248 615 :     SS$_NORMAL      ;SUCCESSFUL COMPLETION
0248 616 :     SS$_ACCVIO      ;ACCESS VIOLATION
0248 617 :     SS$_ILLPAGCNT   ;ILLEGAL PAGE COUNT
0248 618 :     SS$_PAGOWNVIO   ;PAGE OWNER VIOLATION
0248 619 :
0248 620 : SIDE EFFECTS:
0248 621 :     NONE
0248 622 :
0248 623 :--
```



```
0248 625 :
0248 626 : *****
0248 627 :
0248 628 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0248 629 :
0000 0248 630 : .PSECT Y$EXEPAGED
0248 631 :
0248 632 : *****
0248 633 :
00FC 0248 634 : .ENTRY EXE$CNTREG, ^M<R2,R3,R4,R5,R6,R7>
024A 635 :
5E 1C C2 024A 636 : SUBL #MMG$C LENGTH, SP ;RESERVE SCRATCH AREA
57 10 024D 637 : BSBB MMG$RETADRINI ;INIT RETURN RANGE AND SCRATCH AREA
44 50 E9 024F 638 : BLBC R0,30$
51 04 AC 01 C3 0252 639 : SUBL3 #1,PAGCNT(AP),R1 ;R1=DESIRED PAGE COUNT - 1
51 FFE00000 8F D3 0257 640 : BITL #^C<VASM VPG & - ;PAGE COUNT TOO LARGE
025E 641 : ^C<VASM_P1 ! VASM_SYSTEM>>a-9,R1 ;(HIGH 11 BITS NOT CLEAR?)
31 12 025E 642 : BNEQ -20$ ;BRANCH FOR ERROR IF NOT CLEAR
51 51 09 78 0260 643 : ASHL #9,R1,R1 ;FORM BYTE COUNT
0264 644 :
0264 645 : ***** LOCK THE PROCESS HEADER HERE
0264 646 :
0264 647 : SETIPL #IPL$ ASTDEL ;NO AST'S WHILE MODIFYING THE PHD
55 00000000'9F D0 0267 648 : MOVL @#CTL$GL PHD,R5 ;R5=PROCESS HEADER
OF 10 AC E8 026E 649 : BLBS REGION(AP),10$ ;BRANCH IF CONTROL REGION
0272 650 :
0272 651 : FORM R2 = THE FIRST ADDRESS TO DELETE, AND R3 = THE LAST ADDRESS
0272 652 : DELETE STARTING FROM THE END OF THE REGION
0272 653 :
52 28 A5 00000200 8F C3 0272 654 : SUBL3 #^X200,PHD$$_FREPOVA(R5),R2 ;LAST ADR IN P0 SPACE
53 52 51 C3 027B 655 : SUBL3 R1,R2,R3 ;END OF RANGE TO DELETE
0D 11 027F 656 : BRB 15$ ;TO THE COMMON DELETE CODE
52 30 A5 00000200 8F C1 0281 657 10$: ADDL3 #^X200,PHD$$_FREP1VA(R5),R2 ;END OF P1 SPACE
53 52 51 C1 028A 658 : ADDL3 R1,R2,R3 ;END OF RANGE TO DELETE
0091 31 028E 659 15$: BRW DELCOM ;TO THE COMMON DELETE CODE
50 00FC 8F 3C 0291 660 20$: MOVZWL #SS$_ILLPAGCNT,R0 ;ILLEGAL PAGE COUNT
04 0296 661 30$: RET
```

```
0297 663 .SBTTL INADRINI, RETADRINI - INIT SCRATCH AREA AND RANGES
0297 664 :++
0297 665 : FUNCTIONAL DESCRIPTION:
0297 666 :
0297 667 :     GET THE INPUT RANGE, INITIALIZE THE RETURN RANGE, AND INITIALIZE
0297 668 :     THE SCRATCH AREA WHICH IS ASSUMED TO ALREADY BE RESERVED ON THE STACK.
0297 669 :
0297 670 : CALLING SEQUENCE:
0297 671 :
0297 672 :     BSBW    MMG$INADRINI      ;GET INPUT RANGE AND FALL INTO RETADRINI
0297 673 :     BSBW    MMG$RETADRINI    ;INIT RETURN RANGE AND SCRATCH AREA
0297 674 :
0297 675 : INPUT PARAMETERS:
0297 676 :
0297 677 :     NONE
0297 678 :
0297 679 : IMPLICIT INPUTS:
0297 680 :
0297 681 :     MMG$_LENGTH BYTES RESERVED ON TOP OF STACK AND ADDRESSED BY FP
0297 682 :
0297 683 : OUTPUT PARAMETERS:
0297 684 :
0297 685 :     R4, R5 CONTAIN THE INPUT RANGE IF ENTERED AT INADRINI
0297 686 :     MMG$_MAXACMODE(FP) = MAXIMIZED ACCESS MODE
0297 687 :     MMG$_CALLEDIPL(FP) = CALLERS IPL
0297 688 :     MMG$_SAVRETADR(FP) = RETURN ADDRESS SPECIFIED BY USER
0297 689 :     USER SPECIFIED RETURN RANGE INITIALIZED TO -1,-1
0297 690 :
0297 691 : IMPLICIT OUTPUTS:
0297 692 :
0297 693 :     NONE
0297 694 :
0297 695 : COMPLETION CODES:
0297 696 :
0297 697 :     SSS$_ACCVIO      ;ACCESS VIOLATION
0297 698 :
0297 699 : SIDE EFFECTS:
0297 700 :
0297 701 :     NONE
0297 702 :
0297 703 : --
0297 704 :
0297 705 :     MMG$RETRANGE
0297 706 :
0297 707 : ++
0297 708 : FUNCTIONAL DESCRIPTION:
0297 709 :
0297 710 :     THIS ROUTINE RETURNS THE RANGE OF PAGES OPERATED ON TO THE
0297 711 :     USER SPECIFIED ADDRESS. A ZERO ADDRESS INDICATES NO RETURN RANGE
0297 712 :     AND THE ADDRESS IS PROBED FOR WRITING 8 BYTES.
0297 713 :
0297 714 : CALLING SEQUENCE:
0297 715 :
0297 716 :     BSBW    MMG$RETRANGE
0297 717 :
0297 718 : INPUT PARAMETERS:
0297 719 :
```



```

0297 720 :      R1, R2 = RANGE OF ADDRESSES TO BE RETURNED
0297 721 :
0297 722 :      IMPLICIT INPUTS:
0297 723 :
0297 724 :      MMGSL_SAVRETADR(FP) = THE USER SPECIFIED RETURN ADDRESS
0297 725 :
0297 726 :      OUTPUT PARAMETERS:
0297 727 :
0297 728 :      RO PRESERVED
0297 729 :
0297 730 :      IMPLICIT OUTPUTS:
0297 731 :
0297 732 :      NONE
0297 733 :
0297 734 :      COMPLETION CODES:
0297 735 :
0297 736 :      SS$_ACCVIO      ACCESS VIOLATION IF CAN'T WRITE RANGE
0297 737 :
0297 738 :      SIDE EFFECTS:
0297 739 :
0297 740 :      NONE
0297 741 :
0297 742 :--

```

```
0297 744 :
0297 745 : *****
0297 746 :
0297 747 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0297 748 :
00000297 749 : .PSECT Y$EXEPAGED
0297 750 :
0297 751 : *****
0297 752 :
0297 753 MMG$INADRINI::
54 04 AC D0 0297 754      MOVL      INADR(AP),R4      ;GET ADDRESS OF INPUT RANGE
      38 13 0298 755      BEQL      ACCVIORET      ;ERROR IF NO ADDRESS RANGE SPECIFIED
      54 64 7D 029D 756      IFNORD    #8,(R4),ACCVIORET      ;BRANCH IF RANGE IS INACCESSIBLE
      02A3 757      MOVQ      (R4),R4      ;GET THE INPUT RANGE
50 0C AC 02 00 EF 02A6 758 MMG$RETADRINI::
      FD51' 30 02AC 759      EXTZV    #0,#2,ACMODE(AP),R0      ;GET CALLER SPECIFIED ACCESS MODE
      FC AD 50 D0 02AF 760      BSBW    EX$MAXACMODE      ;MAXIMIZE WITH PREVIOUS
      F4 AD 08 AC D0 02B3 761      MOVL    R0,MMG$$_MAXACMODE(FP) ;AND SAVE RESULT IN THE SCRATCH AREA
      51 01 CE 02B7 762      SAVIPL    MMG$$_CALLEDIPL(FP) ;STORE CALLER'S IPL
      52 51 D0 02B8 763      MOVL    RETADR(AP),MMG$$_SAVRETADR(FP) ;GET THE RETURN ADDRESS
      02C2 764      MNEGL    #1,R1      ;INIT THE RANGE TO -1, -1
      02C2 765      MOVL    R1,R2
      02C2 766 :
      02C2 767 : FALL THROUGH TO MMG$RETRANGE
      02C2 768 :
      02C2 769 MMG$RETRANGE::
53  F4 AD D0 02C2 770      MOVL    MMG$$_SAVRETADR(FP),R3 ;USER SPECIFIED RETURN ADR
      09 13 02C6 771      BEQL    20$      ;BRANCH IF NONE SPECIFIED
      63 51 7D 02C8 772      IFNOWRT  #8,(R3),ACCVIORET      ;BRANCH IF NOT WRITABLE
      50 01 D0 02CE 773      MOVQ    R1,(R3)      ;RETURN THE RANGE
      05 05 02D1 774 20$:      MOVL    #SS$_NORMAL,R0
      02D4 775      RSB
      02D5 776 :
      02D5 777 : ACCESS VIOLATION
      02D5 778 :
      02D5 779 :
50 0C 3C 02D5 780 ACCVIORET:
      05 05 02D5 781      MOVZWL  #SS$_ACCVIO,R0      ;SET ACCESS VIOLATION CODE
      02D8 782      RSB      ;AND RETURN IT
```



```
02D9 784 .SBTTL DELTVA - DELETE VIRTUAL ADDRESS SPACE PAGES
02D9 785 :++
02D9 786 : FUNCTIONAL DESCRIPTION:
02D9 787 :
02D9 788 : DELTVA DELETES PAGES STARTING WITH THE SECOND ADDRESS FROM INADR
02D9 789 : ENDING WITH THE FIRST ADR IN INADR. THE SPECIFIED ACCESS MODE
02D9 790 : (MAXIMIZED WITH THE CALLER'S MODE) IS CHECKED AGAINST THE
02D9 791 : OWNER OF EACH PAGE AND MUST BE AT LEAST AS PRIVILEGED.
02D9 792 : EACH PAGE IS MADE INACCESSIBLE.
02D9 793 :
02D9 794 : CALLING SEQUENCE:
02D9 795 :
02D9 796 : CALLG  ARGLIST,@#SYSS$DELTVA
02D9 797 :
02D9 798 : INPUT PARAMETERS:
02D9 800 :
02D9 801 : INADR(AP)      = INPUT RANGE OF ADDRESSES
02D9 802 : RETADR(AP)     = RETURN RANGE OF ADDRESSES
02D9 803 : ACMODE(AP)     = ACCESS MODE TO USE FOR CHECK OF DELETE PRIV
02D9 804 :
02D9 805 : IMPLICIT INPUTS:
02D9 806 : NONE
02D9 807 :
02D9 808 : OUTPUT PARAMETERS:
02D9 809 :
02D9 810 : RO = SYSTEM STATUS CODE
02D9 811 :
02D9 812 : IMPLICIT OUTPUTS:
02D9 813 : NONE
02D9 814 :
02D9 815 : COMPLETION CODES:
02D9 817 :
02D9 818 : SSS_NORMAL      :SUCCESSFUL COMPLETION
02D9 819 : SSS_ACCVIO      :ACCESS VIOLATION
02D9 820 : SSS_PAGOWNVIO   :PAGE OWNER VIOLATION
02D9 821 : SSS_NOPRIV      :NO PRIVILEGE, TRIED TO DELETE SYS PAGE
02D9 822 :
02D9 823 : SIDE EFFECTS:
02D9 824 : NONE
02D9 825 :
02D9 826 :--
```

```
02D9 828 :
02D9 829 : *****
02D9 830 :
02D9 831 : ***** THE FOLLOWING CODE MAY BE PAGED *****
02D9 832 :
0000 02D9 833 : .PSECT Y$EXEPAGED
02D9 834 :
02D9 835 : *****
02D9 836 :
00FC 02D9 837 : .ENTRY EXE$DELTVA,^M<R2,R3,R4,R5,R6,R7>
02DB 838 :
5E 1C C2 02DB 839 : SUBL #MMG$C_LENGTH,SP ;RESERVE SCRATCH AREA
27 B7 10 02DE 840 : BSBB MMG$INADRINI ;GET INPUT RANGE TO R4, R5
50 E9 02E0 841 : BLBC R0,5$
02E3 842 : ;INIT RETURN RANGE AND SCRATCH AREA
52 55 D0 02E3 843 : MOVL R5,R2 ;DELETE FROM THE 2ND ADDRESS
53 54 D0 02E6 844 : MOVL R4,R3 ;THROUGH THE FIRST
02E9 845 :
02E9 846 : ***** LOCK THE PROCESS HEADER HERE
02E9 847 :
02E9 848 : SETIPL #IPL$ ASTDEL ;NO AST'S WHILE MODIFYING THE PHD
55 00000000'9F D0 02EC 849 : MOVL @#CTL$GL_PHD,R5 ;ADDRESS OF PROCESS HEADER
14 52 1E E0 02F3 850 : BBS #VASV_P1,R2,10$ ;BRANCH IF P1 SPACE ADDRESS
02F7 851 :
02F7 852 : P0 SPACE ADDRESS
02F7 853 :
28 A5 52 D1 02F7 854 : CMPL R2,PHD$$_FREPOVA(R5) ;FIRST ADDRESS WITHIN REGION
25 1F 02FB 855 : BLSSU 20$ ;BRANCH IF YES
28 A5 53 D1 02FD 856 : CMPL R3,PHD$$_FREPOVA(R5) ;IF SECOND ADDRESS IS ALSO BEYOND END
1F 1E 0301 857 : BGEQU 20$ ;THEN FORGET ADJUSTING THE RANGE
52 28 A5 01 C3 0303 858 : SUBL3 #1,PHD$$_FREPOVA(R5),R2 ;START WITH FIRST EXISTING PAGE
18 11 0308 859 : BRB 20$
030A 860 :
04 030A 861 : 5$: RET
030B 862 :
030B 863 :
030B 864 : P1 SPACE ADDRESS
030B 865 :
50 30 A5 000001FF 8F C1 030B 866 : 10$: ADDL3 #^X1FF,PHD$$_FREP1VA(R5),R0 ;FIRST BYTE ADR NOT IN P1 SPACE
52 50 D1 0314 867 : CMPL R0,R2 ;FIRST ADDRESS WITHIN REGION
09 1F 0317 868 : BLSSU 20$ ;BRANCH IF YES
53 50 D1 0319 869 : CMPL R0,R3 ;IF SECOND ADDRESS IS ALSO BEYOND END
04 1E 031C 870 : BGEQU 20$ ;THEN FORGET ADJUSTING THE RANGE
52 50 01 C1 031E 871 : ADDL3 #1,R0,R2 ;START WITH FIRST PAGE IN REGION
0322 872 : 20$:
0322 873 :
0322 874 : 0(SP) = SAVED CALLER'S IPL, R2 = STARTING VA, R3 = ENDING VA
0322 875 : R6 = SUBROUTINE TO CALL
0322 876 :
0322 877 : DELCOM:
56 00000141'EF DE 0322 878 : MOVAL L^MMG$DELPAG,R6 ;R6 = ADR OF INITIAL DELETE SCANNER
7E 53 52 C3 0329 879 : ;CALLED TO SCAN WORKING SET LIST
18 10 032D 880 : SUBL3 R2,R3,-(SP) ;SAVE WHETHER STARTVA = ENDVA
8E D5 032F 881 : BSBB MMG$CREDEL ;DO THE DELETION
06 13 0331 882 : TSTL (SP)+ ;IF STARTVA WAS = ENDVA
52 51 7D 0333 883 : BEQL 10$ ;SKIP THE RETURN RANGE EXCHANGE
884 : MOVQ R1,R2 ;EXCHANGE R1 AND R2
```



```
51 53 D0 0336 885      MOVL R3,R1      ;VIA R3
      0339 886 10$:
      0339 887
      50 DD 033D 888      ENBINT MMGSL_CALLEDIPL(FP) ;RESTORE CALLER'S IPL
      81 10 033F 889      PUSHL R0      ;SAVE CURRENT STATUS
02 50 E9 0341 890      BSBB MMG$RETRANGE ;RETURN THE RANGE DELETED
      50 BA 0344 891      BLBC R0,20$ ;ERROR - USE THIS STATUS
      04 0346 892 20$:   POPR R0      ;GET BACK OLD STATUS
      RET                      ;AND EXIT
```

```
0347 894 .SBTTL CREDEL - CREATE/DELETE COMMON LOOP
0347 895
0347 896 : INPUTS
0347 897 : R2 = FIRST ADDRESS TO CREATE/DELETE/SET PROTECTION
0347 898 : R3 = LAST ADDRESS TO CREATE/DELETE/SET PROTECTION
0347 899 : R6 = SUBROUTINE TO CALL
0347 900 : CREPAG, DELPAG, SETPRTPAG, LKWSETPAG, ULWSETPAG
0347 901 : R7 = SAVED AND READY TO USE AS LOOP INCREMENT
0347 902 : R8 = NEW PAGE PROTECTION IF THIS IS SETPRT
0347 903 : MMG$_MAXACMODE(FP) = MAXIMIZED ACCESS MODE
0347 904
0347 905 : OUTPUTS:
0347 906 : R0 = COMPLETION CODE
0347 907 : R1 = FIRST ADDRESS CREATED/DELETED
0347 908 : R2 = LAST ADDRESS CREATED/DELETED
0347 909 : R4 = PCB ADDRESS
0347 910 : R5 = PROCESS HEADER ADDRESS IN P1 SPACE
0347 911 : R6,R7 ALTERED
0347 912
0347 913 : *****
0347 914 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0347 915 : *****
00000347 916 :
0347 917 : .PSECT Y$EXEPAGED
0347 918 : *****
0347 919 :
0347 920 :
0347 921 : MMG$CREDEL::
0347 922 :
0347 923 : EACH OF THE PER PAGE SUBROUTINES IS CALLED WITH THE FOLLOWING REGISTERS
0347 924 : R0 = ACCESS MODE
0347 925 : R2 = VIRTUAL ADDRESS OF PAGE TO OPERATE ON (LOW 9 BITS CLEAR)
0347 926 : R4 = PROCESS CONTROL BLOCK ADDRESS
0347 927 : R5 = PROCESS HEADER ADDRESS (P1 SPACE)
0347 928 : R6 = NUMBER OF PAGES - 1 LEFT TO PROCESS (INCLUDING THE ONE IN R2)
0347 929 : R7 = +^X200 IF GOING FORWARD IN ADDRESS SPACE
0347 930 : = -^X200 IF GOING BACKWARDS IN ADDRESS SPACE
0347 931 :
0347 932 : ALL ROUTINES ARE EXPECTED TO PRESERVE R2, R4, R5, R6, R7, BUT IT IS
0347 933 : OK FOR A ROUTINE TO DO MORE THAN ONE PAGE BY ADDING R7 TO R2 FOR EACH
0347 934 : AND DECREMENTING R6.
0347 935 :
0347 936 :
0347 937 : .ENABL LSB
19 52 1F E0 0347 938 BBS #31,R2,50$ :CHECK FOR SYSTEM SPACE ADDRESS
15 53 1F E0 0348 939 BBS #31,R3,50$ :CHECK FOR SYSTEM SPACE ADDRESS
50 FC AD D0 034F 940 BSBB CRECOM_INIT
50 FC AD D0 0351 941 30$: MOVL MMG$_MAXACMODE(FP),R0 :ACCESS MODE
50 FC AD D0 0355 942 JSB @MMG$_PAGESUBR(FP) :CALL THE PER-PAGE ROUTINE
52 57 C0 0358 943 BLBC R0,40$ :BRANCH IF ERROR CONDITION
52 57 C0 035B 944 ADDL R7,R2 :NEXT VIRTUAL ADDRESS
52 57 C0 035E 945 SOBGEQ R6,30$ :LOOP THROUGH THE PAGES
52 57 C0 0361 946 40$: BSBB CRECOM_DONE
52 57 C0 0363 947 RSB
50 24 3C 0364 948 :
50 24 3C 0364 949 50$: MOVZWL #SS$_NOPRIV,R0 :NO PRIVILEGE
50 24 3C 0364 949 CLRL MMG$_SAVREIADR(FP) :NULL RETURN ADR FOR NULL RANGE
50 24 3C 0367 950
```



```
05 036A 951 RSB
036B 952 .DSABL LSB
036B 953
036B 954 : INITIALIZATION FOR PER/PAGE LOOP
036B 955
036B 956 CRECOM_INIT:
54 00000000'EF D0 036B 957 MOVL L^SCH$GL_CURPCB,R4 ;PROCESS CONTROL BLOCK ADDRESS
55 00000000'9F D0 0372 958 MOVL @^CTL$GL_PHD,R5 ;PROCESS HEADER ADDRESS
FO AD 56 D0 0379 959 MOVL R6,MMG$GL_PAGSUBR(FP) ;ADDRESS OF PER/PAGE SUBROUTINE
57 01 09 78 037D 960 ASHL #9,#1,R7 ;ADDRESS INCREMENT IF GOING FORWARDS
56 53 F7 8F 78 0381 961 ASHL #-9,R3,R6 ;HIGH VIRTUAL PAGE NUMBER
50 52 F7 8F 78 0386 962 ASHL #-9,R2,R0 ;LOW VIRTUAL PAGE NUMBER
56 50 C2 038B 963 SUBL R0,R6 ;NUMBER OF PAGES BASE 0
038E 964
038E 965 : THE FOLLOWING COMPARE MUST BE DONE WITH R2, R3 EXACTLY AS THEY
038E 966 : CAME FROM THE CALLER SO THAT A RANGE OF 205,204 WILL YIELD A RETURN
038E 967 : RANGE OF 3FF,200 INSTEAD OF 200,3FF.
038E 968
53 52 D1 038E 969 CMPL R2,R3 ;GOING FORWARDS IN THE ADDRESS SPACE?
OC 1F 0391 970 BLSSU 10$ ;BRANCH IF YES
04 1A 0393 971 BGTRU 5$ ;BRANCH IF BACKWARDS
06 52 1E E1 0395 972 BBC #VA$V_P1,R2,10$ ;IF EQUAL, THEN FORWARD FOR P0
0399 973 ;BACKWARDS FOR P1
56 56 CE 0399 974 5$: MNEGL R6,R6 ;NO, BACKWARDS, MAKE PAGE COUNT POSITIVE
57 57 CE 039C 975 MNEGL R7,R7 ;ADDRESS INCREMENT = -^X200
52 50 09 78 039F 976 10$: ASHL #9,R0,R2 ;NORMALIZED STARTING VA
EC AD 52 D0 03A3 977 MOVL R2,MMG$GL_SVSTARTVA(FP) ;SAVE STARTING VA
05 03A7 978 RSB
03A8 979
03A8 980 : COMPLETION PROCESSING FOR PER/PAGE LOOP
03A8 981
03A8 982 CRECOM_DONE:
51 EC AD D0 03A8 983 MOVL MMG$GL_SVSTARTVA(FP),R1 ;RECOVER STARTING VIRTUAL ADDRESS
52 51 D1 03AC 984 CMPL R1,R2 ;ANY PAGES SUCCESSFULLY PROCESSED?
48 13 03AF 985 BEQL 60$ ;BRANCH IF NO
52 57 C2 03B1 986 10$: SUBL R7,R2 ;BACK TO LAST PAGE PROCESSED
07 57 1F E0 03B4 987 BBS #31,R7,45$ ;BRANCH IF GOING BACKWARDS
52 01FF 8F A8 03B8 988 BISW #^X1FF,R2 ;ENDVA IS END OF PAGE
05 11 03BD 989 BRB 46$
51 01FF 8F A8 03BF 990 45$: BISW #^X1FF,R1 ;ENDVA IS END OF PAGE
56 56 0080 C4 D0 03C4 991 46$: MOVL PCB$GL_JIB(R4),R6
56 38 A6 3C A6 C3 03C9 992 SUBL3 JIB$GL_PGFLCNT(R6),JIB$GL_PGFLQUOTA(R6),R6 ;PAGE FILE PAGES IN USE
00000000'GF 56 D1 03CF 993 CMPL R6,G^CTL$GL_VIRTPEAK ;MAINTAIN MAX PAGE FILE USED
10 1F 03D6 994 BLSSU 50$
00000000'GF 56 D0 03D8 995 MOVL R6,G^CTL$GL_VIRTPEAK ;SET NEW PEAK VALUE
00000000'GF 56 D0 03DF 996 MOVL R6,G^CTL$GL_IPAGEFL ;SET NEW PEAK VALUE
10 11 03E6 997 BRB 52$
00000000'GF 56 D1 03E8 998 50$: CMPL R6,G^CTL$GL_IPAGEFL ;MAINTAIN MAX PAGE FILE USED (IMAGE)
07 1F 03EF 999 BLSSU 52$
00000000'GF 56 D0 03F1 1000 MOVL R6,G^CTL$GL_IPAGEFL ;SET NEW PEAK VALUE
05 03F8 1001 52$: RSB
03F9 1002
2C 50 D1 03F9 1003 60$: CMPL R0,#SS$ _ABORT ;WAS IT AN ABORT OF A CRMPSC
B3 13 03FC 1004 BEQL 10$ ;YES - FIX UP R2, LEAVE RETADR ALONE
F4 AD D4 03FE 1005 CLRL MMG$GL_SAVRETADR(FP) ;NULL RETURN ADR FOR NULL RANGE
05 0401 1006 RSB
0402 1007
```



```
0402 1009      .SBTTL  CREPAG - CREATE A SINGLE PAGE
0402 1010
0402 1011      :++
0402 1012      : FUNCTIONAL DESCRIPTION:
0402 1013      :   CREPAG CREATES ONE PAGE IN THE P0 OR P1 PAGE TABLE.  IF NECESSARY
0402 1014      :   THE PAGE TABLE IS EXTENDED, WHICH CAN FAIL WHEN THERE ARE NO PTE'S
0402 1015      :   AVAILABLE.  THE PTE IS DELETED IF NECESSARY AND IS THEN SET
0402 1016      :   TO THE VALUE SPECIFIED.
0402 1017
0402 1018      : CALLING SEQUENCE:
0402 1019
0402 1020      :       BSBW      MMG$CREPAG
0402 1021
0402 1022      : INPUT PARAMETERS:
0402 1023
0402 1024      :   R0 = MODE FOR CREATING NEW PAGE IN LOW BYTE
0402 1025      :       MMG$V CHGPAGFIL SET IF CHARGING PAGE FILE QUOTA FOR THIS PAGE
0402 1026      :   R2 = VIRTUAL ADDRESS OF PAGE TO CREATE (LOW 9 BITS = 0)
0402 1027      :   R4 = PCB ADDRESS
0402 1028      :   R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
0402 1029      :   R6 = COUNT - 1 OF PAGES REMAINING TO PROCESS INCLUDING THIS ONE
0402 1030      :   R7 = +^X200 IF GOING FORWARDS, -^X200 IF GOING BACKWARDS
0402 1031      :   R8 = NEW CONTENTS OF PAGE TABLE ENTRY
0402 1032
0402 1033      :   THE CURRENT IPL MUST BE AT ASTDEL
0402 1034
0402 1035      : IMPLICIT INPUTS:
0402 1036      :   NONE
0402 1037
0402 1038      : OUTPUT PARAMETERS:
0402 1039
0402 1040      :   R0 = ERROR STATUS CODE
0402 1041      :   R2 PRESERVED
0402 1042
0402 1043      : IMPLICIT OUTPUTS:
0402 1044
0402 1045      :   PTE CORRESPONDING TO SPECIFIED VIRTUAL ADDRESS IS DELETED AND
0402 1046      :   THE DESIRED PTE IS STORED
0402 1047
0402 1048      :   IF PAGE TABLE EXPANSION IS NECESSARY THEN THE FOLLOWING
0402 1049      :   ARE AFFECTED:
0402 1050
0402 1051      :   PHD$$_FREPOVA OR PHD$$_FREPIVA      :1ST FREE PAGE AT END OF P0/P1 PAGE TABLE
0402 1052      :   PHD$$_POLRASTL OR PHD$$_P1LR      :LENGTH OF PT IN HARDWARE PCB
0402 1053      :   PR$$_POLR OR PR$$_P1LR              :LENGTH OF PT IN PROCESSOR REG
0402 1054      :   PHD$$_FREPTCNT                      :FREE PTE COUNTER
0402 1055
0402 1056      : COMPLETION CODES:
0402 1057
0402 1058      :   $$$_NORMAL                          :SUCCESSFUL COMPLETION
0402 1059      :   $$$_VASFULL                          :VIRTUAL ADDRESS SPACE FULL
0402 1060
0402 1061      : SIDE EFFECTS:
0402 1062      :   NONE
0402 1063
0402 1064      :--
```



```
0402 1066 :
0402 1067 : *****
0402 1068 :
0402 1069 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0402 1070 :
0000 0402 1071 : .PSECT Y$EXEPAGED
0402 1072 :
0402 1073 : *****
0402 1074 :
0402 1075 : .ENABL LSB
0402 1076 :
0402 1077 :
0402 1078 : PAGE TABLE ENTRY IS NOT EMPTY, MUST DELETE THE PAGE FIRST
0402 1079 :
0402 1080 : DELETED FIRST:
50 04 AE 9A 0402 1081 : MOVZBL 4(SP),R0 : ACCESS MODE
00000141'EF 16 0406 1082 : JSB MMG$DELPAG : DELETE THE PAGE
53 50 E9 040C 1083 : BLBC R0,35$ : BRANCH IF ERROR
50 03 BA 040F 1084 : POPR #^M<R0,R1> : R1 = ACCESS MODE
50 51 DO 0411 1085 : MOVL R1,R0 : PLACE IT IN R0
0414 1086 :
0414 1087 : FALL THROUGH TO CREPAG
0414 1088 :
0414 1089 : MMG$CREPAG::
17 52 50 DD 0414 1090 : PUSHL R0 : SAVE ACCESS MODE
1E E1 0416 1091 : BBC #VAV_P1,R2,20$ : BRANCH IF PO SPACE
041A 1092 :
041A 1093 : P1 SPACE
041A 1094 :
7E 53 00D0 C5 DE 041A 1095 : MOVAL PHD$L_P1BR(R5),R3 : ADR OF POINTER TO P1PT
30 A5 52 C3 041F 1096 : SUBL3 R2,PHD$L_FREP1VA(R5),-(SP) : BEYOND END OF REGION?
4D 19 0424 1097 : BLSS 80$ : BRANCH IF NOT
0426 1098 :
0426 1099 : 0(SP) = NO. OF BYTES BEYOND P1VA (INTEGRAL NO. OF PAGES)
0426 1100 : NOTE THAT THIS IS ^X200 SHORT OF WHAT MUST BE ADDED TO REGION
0426 1101 :
51 56 57 C5 0426 1102 : MULL3 R7,R6,R1 : NO. OF BYTES LEFT TO PROCESS
042A 1103 : : + IF GOING FORWARDS, - IF GOING BACKWARDS
39 18 042A 1104 : BGEQ 40$ : BRANCH IF EXPANSION IS NOT IN THE
042C 1105 : : NATURAL GROWTH DIRECTION OF THE REGION
51 51 CE 042C 1106 : MNEGL R1,R1 : NO. OF BYTES TO ADD TO SIZE OF REGION
12 11 042F 1107 : BRB 30$ : TRY TO EXPAND TO FIT LAST PAGE
0431 1108 :
0431 1109 : PO SPACE
0431 1110 :
7E 53 00C8 C5 DE 0431 1111 20$: MOVAL PHD$L_POBR(R5),R3 : ADR OF POINTER TO POPT
52 28 A5 C3 0436 1112 : SUBL3 PHD$L_FREPOVA(R5),R2,-(SP) : BEYOND END OF REGION?
36 19 043B 1113 : BLSS 80$ : BRANCH IF NOT
043D 1114 :
043D 1115 : 0(SP) = NO. OF BYTES BEYOND POVA (INTEGRAL NO. OF PAGES)
043D 1116 : NOTE THAT THIS IS ^X200 SHORT OF WHAT MUST BE ADDED TO REGION
043D 1117 :
51 56 57 C5 043D 1118 : MULL3 R7,R6,R1 : NO. OF BYTE LEFT TO BE CREATED
0441 1119 : : + IF GOING FORWARD, - IF GOING BACKWARDS
22 15 0441 1120 : BLEQ 40$ : BRANCH IF NOT FORWARD
0443 1121 :
0443 1122 : TRY EXPANDING REGION TO FIT LAST PAGE TO BE CREATED
```



```
0443 1123 : CREATION IS IN THE DIRECTION OF THE EXPANDING REGION
0443 1124 :
51 51 6E C0 0443 1125 30$: ADDL (SP),R1 ;AMOUNT EVENTUALLY NEEDED TO EXPAND BY
0200 C1 DE 0446 1126 MOVAL ^X200(R1),R1 ;
0084 30 044B 1127 BSBW EXPANDCHK ;TRY EXPANDING BY THAT MUCH
22 50 E8 044E 1128 BLBS R0,80$ ;BRANCH IF REGION EXPANDED
0244 8F 50 B1 0451 1129 CMPW R0,#SS$_VASFULL
OD 12 0456 1130 BNEQ 40$ ;NOT VIRTUAL ADDRESS SPACE FULL
0458 1131 :
0458 1132 :
0458 1133 : CAN'T EXPAND REGION BY THE FULL AMOUNT THAT WILL BE NEEDED
0458 1134 : SO EXPAND BY WHAT IS LEFT UNLESS THAT'S NOT ENOUGH FOR THE CURRENT PAGE
0458 1135 :
51 2C A5 09 9C 0458 1136 ROTL #9,PHD$_FREPTCNT(R5),R1 ;SPACE LEFT FOR EXPANDING
6E 51 D1 045D 1137 CMPL R1,(SP) ;EXPAND BY SPACE REMAINING UNLESS
0B 14 0460 1138 ;MORE IS NEEDED FOR THIS PAGE
005C 31 0460 1139 BGTR 50$ ;BRANCH IF THERE'S ENOUGH
0462 1140 35$: BRW 90$ ;OTHERWISE VIRTUAL ADR SPACE FULL
0465 1141 :
0465 1142 : JUST EXPAND FOR THIS PAGE
0465 1143 :
51 6E 00000200 8F C1 0465 1144 40$: ADDL3 #^X200,(SP),R1 ;AMOUNT NEEDED FOR THIS PAGE
0062 30 046D 1145 50$: BSBW EXPANDCHK ;TRY EXPANDING BY THIS AMOUNT
4E 50 E9 0470 1146 BLBC R0,90$ ;BRANCH IF NO ROOM
0473 1147 :
0473 1148 : 0(SP) = GARBAGE, 4(SP) = ACCESS MODE, 5(SP) NONZERO IF CHARGING PAGE FILE QUOTA
0473 1149 : R2 = VA, R3 = ADDRESS OF POINTER TO APPROPRIATE PAGE TABLE
0473 1150 :
51 52 15 09 EF 0473 1151 80$: EXTZV #VAS$_VPN,#VASS$_VPN,R2,R1 ;VIRTUAL PAGE NUMBER
00 B341 D5 0478 1152 TSTL @ (R3)[R1] ;PTE EMPTY?
84 12 047C 1153 BNEQ DELETEDFIRST ;BRANCH IF NO, MUST DELETE IT
0A 04 AE 08 E1 047E 1154 BBC #MMG$_CHGPAGFIL,4(SP),85$ ;BRANCH IF NOT CHARGING FOR PAGE FILE
50 0080 C4 D0 0483 1155 MOVL PCB$_JIB(R4),R0 ;GET JIB ADDRESS
3C A0 D7 0488 1156 DECL JIB$_PGFLCNT(R0) ;CHARGE A PAGE FILE PAGE
38 19 048B 1157 BLSS 100$ ;BRANCH IF OVER QUOTA
048D 1158 :
00 B341 58 D0 048D 1159 85$: MOVL R8,@ (R3)[R1] ;STORE NEW PTE
0492 1160 :
28 58 1F E1 0492 1161 BBC #PTES$_VALID,R8,87$ ;BR IF NOT WINDOW/MA780 GLOBAL PAGE
OE 58 15 E0 0496 1162 BBS #PTES$_WINDOW,R8,86$ ;BR IF IT IS A WINDOW PAGE
7E 58 15 00 EF 049A 1163 EXTZV #PTES$_PFN,#PTES$_PFN,R8,-(SP) ;EXTRACT PFN
00000000'GF 8E D1 049F 1164 CMPL (SP)+,G^MMG$_GL_MAXMEM ;IS THIS A MA780 GLOBAL PAGE?
16 15 04A6 1165 BLEQ 87$ ;BR IF NOT MA780 GLOBAL PAGE
53 9341 DE 04A8 1166 86$: MOVAL @ (R3)+[R1],R3 ;GET SVAPTE
50 01 D0 04AC 1167 MOVL #1,R0 ;INDICATE ADDITIONAL LOCK
00000004'GF 16 04AF 1168 DSBINT #IPL$_SYNCH ;RAISE TO SYNCH
04B5 1169 JSB G^MMG$_MOVPTLOCK ;LOCK THE PAGE TABLE PAGE IN WS
04BB 1170 ENBINT ;RESTORE IPL
04BE 1171 :
50 01 3C 04BE 1172 87$: MOVZWL #SS$_NORMAL,R0 ;INDICATE SUCCESSFUL COMPLETION
5E 08 C0 04C1 1173 90$: ADDL #2*4,SP ;CLEAN OFF STACK
05 04C4 1174 RSB
04C5 1175 :
04C5 1176 : EXCEEDED PAGE FILE QUOTA
04C5 1177 :
50 0080 C4 D0 04C5 1178 100$: MOVL PCB$_JIB(R4),R0 ;GET JIB ADDRESS
3C A0 D6 04CA 1179 INCL JIB$_PGFLCNT(R0) ;RESTORE CORRECT COUNT
```


SYSCREDEL
V04-000

- SYSTEM SERVICE CREATE & DELETE PAGE
CREPAG - CREATE A SINGLE PAGE

F 12

16-SEP-1984 01:49:03 VAX/VMS Macro V04-00
5-SEP-1984 03:49:40 [SYS.SRC]SYSCREDEL.MAR;1

Page 26
(16)

50	1C	3C	04CD	1180	MOVZWL	#SS\$ _EXQUOTA,R0	;EXCEEDED QUOTA
	EF	11	04D0	1181	BRB	90\$;RETURN THE ERROR
			04D2	1182			
			04D2	1183	.DSABL	LSB	

```
04D2 1185 .SBTTL MMG$MOVPTLOCK - Move WSLE for PT into locked portion of WS
04D2 1186
04D2 1187 :
04D2 1188 : FUNCTIONAL DESCRIPTION:
04D2 1189 :
04D2 1190 : This routine is used to lock in the working set those page table pages
04D2 1191 : that contain WINDOW PTEs and PTEs mapping MA780 global section pages.
04D2 1192 : This is needed because PFNMAP and MA780 pages do not have working set
04D2 1193 : list entries, which would allow the user to control locking in the
04D2 1194 : working set.
04D2 1195 :
04D2 1196 : The page table pages are moved into the locked portion of the working
04D2 1197 : set, instead of just turning on the locked bit in the WSLE for two
04D2 1198 : reasons. First, there would be no way to outswap the process header if
04D2 1199 : the normal locking was done by incrementing the PFN database SHRCNT,
04D2 1200 : unless the swapper was taught about the PTWSLELCK array. Second,
04D2 1201 : moving a WSLE that cannot be replaced, out of the way of the working
04D2 1202 : set scan routine, optimizes the scan -- as is done for normal P0/P1
04D2 1203 : pages that an user locks in the working set.
04D2 1204 :
04D2 1205 : INPUTS:
04D2 1206 :
04D2 1207 : R0 = Indicator of whether to increment or decrement the
04D2 1208 : page table page WSLE lock array (+1 or -1)
04D2 1209 : R3 = SVAPTE for some virtual address in P0 or P1
04D2 1210 : R4 = PCB of process
04D2 1211 : R5 = PHD of process
04D2 1212 :
04D2 1213 : IPL = SYNCH
04D2 1214 :
04D2 1215 : For MMG$MOVPTLOCK1 only:
04D2 1216 :
04D2 1217 : R2 = Adr of count byte for # locked WSLEs in this page table page
04D2 1218 :
04D2 1219 : OUTPUTS:
04D2 1220 :
04D2 1221 : All registers unchanged.
04D2 1222 :
04D2 1223 : The working set list entry for the page table page that contains
04D2 1224 : the PTE pointed to by R3, may be moved in or out of the locked portion
04D2 1225 : of the working set. The page table page WSLE lock array is altered as
04D2 1226 : appropriate as is the count of locked page table pages.
04D2 1227 :
00000000 1228 .PSECT $MMGCODE
0000 1229 .ENABL LSB
0000 1230 MMG$MOVPTLOCK1::
OF BB 0000 1231 PUSH R0,R1,R2,R3 ;Save registers
15 11 0002 1232 BRB 10$ ;Continue in common code
0004 1233 MMG$MOVPTLOCK:: ;Called for PFNMAP/MA780 global pages
OF BB 0004 1234 PUSH R0,R1,R2,R3 ;Save registers
0006 1235
52 53 00C8 C5 C3 0006 1236 SUBL3 PHD$ POBR(R5),R3,R2 ;Byte offset of PTE
51 52 F7 8F 78 000C 1237 ASHL #-9,R2,R1 ;Byte index of containing page table
52 55 64 A5 C1 0011 1238 ADDL3 PHD$ PTWSLELCK(R5),R5,R2 ;Locked working set list entries
52 51 C0 0016 1239 ADDL2 R1,R2 ;Address of count byte for # of locked
0019 1240 ; WSLE's in the page table
0019 1241 10$:
```



```
50 D5 0019 1242 TSTL R0 ;Is this an increment or decrement?
09 19 001B 1243 BLSS 20$ ;Br if it is a decrement
62 96 001D 1244 INCB (R2) ;Add one for WINDOW/MA780 global page
0C 14 001F 1245 BGTR 40$ ;Branch if not the first in this PT
6C A5 B6 0021 1246 INCW PHD$W_PTCNTLCK(R5) ;Another page table with locked WSLE's
07 11 0024 1247 BRB 40$ ;Continue in common code
62 97 0026 1248 20$: DECB (R2) ;One less reason to lock page table page
42 18 0028 1249 BGEQ 100$ ;Br if PT page should remain locked
6C A5 B7 002A 1250 DECW PHD$W_PTCNTLCK(R5) ;One less page table with locked WSLE's
002D 1251 40$:
002D 1252
002D 1253 ; Must move page table page into/out of locked portion of working set.
002D 1254 ; The WSL$M_WSLOCK bit is not used for locking because there is no count
002D 1255 ; that tells the system when it can be cleared.
002D 1256
51 52 0E A5 3C 002D 1257 MOVZWL PHD$W_WSDYN(R5),R2 ;Index to second working set list slot
53 15 09 EF 0031 1258 EXTZV #VAS$-VPN,#VASS_VPN,R3,R1 ;Page # of PT containing this PTE
50 51 0000'DF41 D0 0036 1259 MOVL @W^MMG$GL_SPTBASE[R1],R1 ;PTE for this page table page
51 51 15 00 EF 003C 1260 EXTZV #PTES$V_PFN,#PTES$S_PFN,R1,R0 ;PFN for this page table page
0041 1261 .list MEB
0041 1262 PFN REFERENCE - ;Get the WSLX
0041 1263 MOVZWL <@W^PFN$Ax_WSLX[R0],R1>,-
0041 1264 LONG_OPCODE=MOVL,-
0041 1265 IMAGE=SYS_NONPAGED
00000000 .SAVE PSECT LOCAL_BLOCK
00000041' 0000 .PSECT Z$INIT$PFN_FIXUP_TABLE
00000041' 0000 .ADDRESS ...PFN
3C 0004 .BYTE OPS_MOVZWL
D0 0005 .BYTE OPS_MOVL
00000041 .RESTORE PSECT
51 0000'DF40 3C 0041 MOVZWL @W^PFN$Ax_WSLX[R0],R1
6E D5 0047 1266 TSTL (SP) ;Is this an lock or an unlock?
07 19 0049 1267 BLSS 80$ ;Br if it is an unlock
52 51 D1 004B 1268 CMPL R1,R2 ;Is WSLE already in locked portion?
1C 19 004E 1269 BLSS 100$ ;Br if already there
0D 11 0050 1270 BRB 90$ ;Go swap WSLE into locked portion
52 51 D1 0052 1271 80$: CMPL R1,R2 ;Is WSLE in locked portion?
15 18 0055 1272 BGEQ 100$ ;Br if in dynamic portion
OC A5 51 B1 0057 1273 CMPW R1,PHD$W_WSLOCK(R5) ;Is it in permanent part of WS
OF 1F 005B 1274 BLSSU 100$ ;Yes - leave it there
52 D7 005D 1275 DECL R2 ;Swapping WSLE with last locked entry
005F 1276
005F 1277 ; R0 = PFN of first WSL slot
005F 1278 ; R1 = Index to first WSL slot
005F 1279 ; R2 = Index to second WSL slot
005F 1280 ; R4 = PCB address
005F 1281 ; R5 = PHD address - P1 or system space
005F 1282
005F 1283 ; IPL = SYNCH
005F 1284
00000000'EF 16 005F 1285 90$: JSB MMG$SWAPWSLE
OE A5 6E A0 0065 1286 ADDW (SP),PHD$W_WSDYN(R5) ;Adjust last locked slot in dynamic WS
FF94' 30 0069 1287 BSBW MMG$EXTRADYNWS ;Re-compute extra dynamic working set
OF BA 006C 1288 100$: POPR #*M<R0,R1,R2,R3> ;Restore registers
05 006E 1289 RSB ;PT is now in locked portion of WS
006F 1290
006F 1291 .DSABL LSB
```

```
006F 1293      .SBTTL EXPANDCHK - EXPAND REGION CHECKING FOR SPACE AVAILABLE
006F 1294
006F 1295      :
006F 1296      : INPUTS:
006F 1297      :
006F 1298      : R1 = NO. OF BYTES TO EXPAND BY (INTEGRAL NO. OF PAGES)
006F 1299      : R2 = VIRTUAL ADDRESS (PROVIDING THE PO/P1 SPACE INDICATOR)
006F 1300      : R5 = PROCESS HEADER ADDRESS
006F 1301      :
006F 1302      : OUTPUTS:
006F 1303      :
006F 1304      : R0 = STATUS CODE
006F 1305      : SSS_NORMAL FOR SUCCESSFUL
006F 1306      : SSS_VASFULL FOR VIRTUAL ADDRESS SPACE FULL
006F 1307      : SSS_INSFWSL FOR INSUFFICIENT WORKING SET LIMIT
006F 1308      : R2, R3 PRESERVED
006F 1309
006F 1310
006F 1311      : *****
006F 1312      : ***** THE FOLLOWING CODE MAY BE PAGED *****
006F 1313      : *****
006F 1314      :
0000 04D2 1315      .PSECT Y$EXEPAGED
04D2 1316
04D2 1317      : *****
04D2 1318
04D2 1319 EXPANDCHK:
50 51 F7 8F 78 04D2 1320      ASHL    #-9,R1,R0      ;NUMBER OF PAGES REQUIRED
2C A5 50 C2 04D7 1321      SUBL    R0,PHD$L_FREPTCNT(R5) ;ENOUGH FREE PAGE TABLE ENTRIES
10 52 2C 19 04DB 1322      BLSS    40$      ;BRANCH IF NOT
E0 04DD 1323      BBS     #VASV_P1,R2,20$ ;BRANCH IF P1 SPACE ADDRESS
04E1 1324      :
04E1 1325      : PO SPACE ADDRESS
04E1 1326      :
00CC C5 50 C0 04E1 1327      ADDL    R0,PHD$L_POLRASTL(R5) ;ADJUST LENGTH OF REGION
28 A5 51 C0 04E6 1328      ADDL    R1,PHD$L_FREPOVA(R5) ;AND NEXT FREE PO VIRTUAL ADDRESS
09 00CC C5 DA 04EA 1329      LDPOLR  PHD$L_PO[RASTL(R5) ;ADJUST HARDWARE LENGTH REGISTER
OE 11 04EF 1330      MTPR    PHD$L_POLRASTL(R5),S^#PR$ POLR
04F1 1331      BRB     30$      ;EXIT THROUGH COMMON CODE
04F1 1332      :
04F1 1333      : P1 SPACE ADDRESS
00D4 C5 50 C2 04F1 1334 20$: SUBL    R0,PHD$L_P1LR(R5) ;ADJUST LENGTH OF REGION
30 A5 51 C2 04F6 1335      SUBL    R1,PHD$L_FREP1VA(R5) ;AND NEXT FREE P1 VIRTUAL ADR
OB 00D4 C5 DA 04FA 1336      LDP1LR  PHD$L_P1[CR(R5) ;AND HARDWARE LENGTH REGISTER
48 10 04FF 1337 30$: BSBB    UPDATPTCNT ;UPDATE MAX PAGE TABLE COUNT
51 B5 0501 1338      TSTW    R1 ;IF NOT ENOUGH DYNAMIC WSLE'S LEFT
OE 13 0503 1339      BEQL    50$ ;THEN TOO MUCH VA SPACE FOR WS LIMIT
50 01 3C 0505 1340      MOVZWL  #SS$_NORMAL,R0 ;SUCCESSFUL COMPLETION CODE
05 0508 1341      RSB
0509 1342      :
0509 1343      : NOT ENOUGH ROOM IN VIRTUAL ADDRESS SPACE FOR SPECIFIED EXPANSION
0509 1344      :
2C A5 50 C0 0509 1345 40$: ADDL    R0,PHD$L_FREPTCNT(R5) ;FIX UP FREE PAGE TABLE ENTRY COUNTER
50 0244 8F 3C 050D 1346      MOVZWL  #SS$_VASFULL,R0 ;RETURN 'VIRTUAL ADDRESS SPACE FULL'
05 0512 1347      RSB
```



```
0513 1348 :  
0513 1349 : ADDITION OF DESIRED VIRTUAL ADDRESS SPACE WILL RESULT IN TOO FEW  
0513 1350 : DYNAMIC PAGES IN THE WORKING SET LIST. REFUSE TO ADD THE SPACE UNLESS  
0513 1351 : WE ARE BELOW QUOTA.  
0513 1352 :  
50 18 A5 08 A5 A3 0513 1353 50$: SUBW3 PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R0  
50 A5 50 B6 0519 1354 INCW R0 ;GET QUOTA  
50 1B 1B 051B 1355 CMPW R0,PHD$W_WSSIZE(R5) ;ARE WE BELOW QUOTA  
50 51 DD 051F 1356 BLEQU 55$ ;NO  
7E 50 A5 3C 0521 1357 PUSHL R1 ;SAVE VOLIATILE REGISTER  
7E 7F 8F 3C 0523 1358 MOVZWL PHD$W_WSSIZE(R5),-(SP) ;SAVE CURRENT WSSIZE  
00000000'EF 02 DD 0527 1359 PUSHL #0 ;NO ADDRESS FOR PREVIOUS VALUE  
50 50 A5 9A 0529 1360 MOVZBL #127, -(SP) ;AUGMENT WS BY 127 PAGES  
50 50 A5 03 FB 052D 1361 CALLS #2, EXE$ADJWSL ;CALL INTERNAL ADJUST WS ROUTINE  
50 50 A5 03 BA 0534 1362 POPR #^M<R0,R1> ;RESTORE SAVED WSSIZE, R1  
50 50 A5 12 0536 1363 CMPW PHD$W_WSSIZE(R5),R0 ;DID WS GROW AT ALL?  
50 50 A5 C3 053A 1364 BNEQ 30$ ;YES, GO TRY AGAIN  
50 50 A5 53 DD 053C 1365 55$: PUSHL R3  
50 002C 30 053E 1366 BSBW MMG$CONTRACT ;GO CLEAN UP  
50 011C 8F 08 BA 0541 1367 POPR #^M<R3>  
50 011C 8F 3C 0543 1368 MOVZWL #SS$_INSFWSL,R0 ;INSUFFICIENT WORKING SET LIMIT  
05 0548 1369 RSB
```

```
0549 1371      .SBTTL UPDATPTCNT - UPDATE (MAX) PAGE TABLE COUNT
0549 1372
0549 1373      : FUNCTIONAL DESCRIPTION:
0549 1374      :
0549 1375      :     THIS ROUTINE CALCULATES THE MAXIMUM NUMBER OF PAGE TABLES THAT
0549 1376      :     MAY BE REQUIRED TO SUPPORT THE VIRTUAL ADDRESS SPACE CURRENTLY CREATED.
0549 1377      :     THE CALCULATION IS BASED SOLELY ON THE CURRENT SETTINGS FOR THE ENDS
0549 1378      :     OF P0 AND P1 SPACE, HOLES IN THE ADDRESS SPACE ARE NOT DEALT WITH.
0549 1379      :
0549 1380      : INPUTS:
0549 1381      :
0549 1382      :     R5 = PROCESS HEADER ADDRESS (P1 SPACE IF PROCESS)
0549 1383      :
0549 1384      : OUTPUTS:
0549 1385      :
0549 1386      :     R1 = WORD CONTAINING EXTRA DYNAMIC WORKING SET LIST ENTRY COUNT
0549 1387      :     R2,R3 PRESERVED
0549 1388      :
0549 1389      : *****
0549 1390      : ***** THE FOLLOWING CODE MAY BE PAGED *****
0549 1391      :
0549 1392      :
00000549 1393      .PSECT Y$EXEPAGED
0549 1394      :
0549 1395      : *****
0549 1396      :
0549 1397      UPDATPTCNT:
0549 1398      EXTZV  #0,#7,PHD$ POLRASTL(R5),R0 ;NO. OF ENTRIES TO FILL LAST POPT
0550 1399      SUBL3  R0,PHD$ _FREPTCNT(R5),R0 ;NO. OF FREE PTE'S STARTING
0555 1400      :AT THE NEXT PAGE TABLE BOUNDARY
0555 1401      ASHL   #-7,R0,R0 ;TRUNCATED DIVIDE BY 128 YIELDS
055A 1402      :COUNT OF EMPTY PAGE TABLES
055A 1403      BGEQ   10$ ;BRANCH IF SOME EMPTY PAGE TABLES
055C 1404      CLRL   R0 ;NO EMPTY PAGE TABLES
055E 1405      SUBW3  R0,SGN$GL PTPAGCNT,PHD$W PTCNTMAX(R5) ;SET MAX PAGE TABLE COUNT
0567 1406      JMP    MMG$EXTRADYNWS ;COMPUTE AND RETURN EXTRA DYNAMIC WSLE COUNT
```

50 00CC C5 07 00 EF 0549 1398
50 2C A5 50 C3 0550 1399
50 50 F9 8F 78 0555 1400
02 18 055A 1402
50 D4 055A 1403
72 A5 00000000'EF 50 A3 055C 1404
00000000'EF 17 055E 1405 10\$:
0567 1406


```
056D 1408 .SBTTL DELPAG - DELETE A SINGLE PAGE
056D 1409 :++
056D 1410 : FUNCTIONAL DESCRIPTION:
056D 1411 :
056D 1412 :     THIS ROUTINE DELETES THE PAGE SPECIFIED, CLEARING ITS PAGE TABLE
056D 1413 :     ENTRY. IF THIS IS THE LAST PAGE OF THE RANGE TO BE DELETED OR IF
056D 1414 :     IT RESULTS IN A LENGTH VIOLATION, THEN THE CONTRACT REGION LOGIC IS CALLED.
056D 1415 :
056D 1416 : CALLING SEQUENCE:
056D 1417 :
056D 1418 :     BSBW    MMG$DELPAG
056D 1419 :
056D 1420 :
056D 1421 : INPUT PARAMETERS:
056D 1422 :
056D 1423 :     R0 = ACCESS MODE TO CHECK AGAINST PAGE OWNER
056D 1424 :     MMG$V_DELGBLON SET IF ALREADY HAVE DONE A PURGE OF THE GLOBAL
056D 1425 :     PAGES IN THE SPECIFIED ADDRESS RANGE
056D 1426 :     MMG$V_NOWAIT_IPLO SET TO RETURN SSS_ABORT RATHER THAN WAITING
056D 1427 :     FOR I/O COMPLETION AT IPL 0
056D 1428 :     R2 = VIRTUAL ADDRESS (LOW 9 BITS = 0)
056D 1429 :     R4 = PCB ADDRESS
056D 1430 :     R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
056D 1431 :     R6 = COUNT - 1 OF PAGES REMAINING TO BE DELETED INCLUDING THIS ONE
056D 1432 :     R7 = +^X200 IF GOING FORWARD IN ADDRESS SPACE
056D 1433 :     = -^X200 IF GOING BACKWARDS IN ADDRESS SPACE
056D 1434 :     IPL = ASTLVL
056D 1435 :
056D 1436 : IMPLICIT INPUTS:
056D 1437 :     NONE
056D 1438 :
056D 1439 : OUTPUT PARAMETERS:
056D 1440 :
056D 1441 :     R0 = STATUS CODE
056D 1442 :     R2 PRESERVED
056D 1443 :
056D 1444 : IMPLICIT OUTPUTS:
056D 1445 :     NONE
056D 1446 :
056D 1447 : COMPLETION CODES:
056D 1448 :
056D 1449 :     SSS_NORMAL                ;SUCCESSFUL RETURN
056D 1450 :     SSS_PAGOWNVIO            ;PAGE OWNER VIOLATION
056D 1451 :
056D 1452 : SIDE EFFECTS:
056D 1453 :
056D 1454 :     R1,R3 - DESTROYED
056D 1455 :
056D 1456 :     CAUTION - MAY WAIT AT IPL 0
056D 1457 :
056D 1458 :--
```

```
056D 1460 :
056D 1461 : *****
056D 1462 :
056D 1463 : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
056D 1464 :
0000006F 1465 : .PSECT $MMGCODE
006F 1466 :
006F 1467 : *****
006F 1468 :
006F 1469 :
006F 1470 : DELETE A PAGE MAPPED BY PFN
006F 1471 :
006F 1472 : PFNMAPDEL:
006F 1473 : INVALID R2 ;INVALIDATE THE TRANSLATION BUFFER
3E A4 40 A4 B1 0072 1474 : CMPW PCBSW_DIOLM(R4),PCBSW_DIOCNT(R4) ;ANY DIRECT I/O FOR PROCESS?
03 1B 0077 1475 : BLEQU 10$ ;BR ON NO, NO I/O COMPLETION TO WAIT FOR
0188 31 0079 1476 : BRW DELPAGASTWAIT ;GO WAIT FOR I/O TO COMPLETE
16 50 15 E5 007C 1477 10$: BBCC #PTESV_WINDOW,R0,SHMPAGDEL ;BR IF SHARED MEMORY GLOBAL PAGE
00000000'EF 50 D1 0080 1478 : CMPL R0,MMG$GL_MAXMEM ;IS THE PAGE SHARED BY BOTH CPU'S OF MP
04 1B 0087 1479 : BLEQU DELPTE1 ;YES - DON'T DECREMENT INHIBIT COUNT
0104 C5 D7 0089 1480 : DECL PHDSL_MPINHIBIT(R5) ;REMOVE ONE REASON THAT PROCESS MUST
008D 1481 : ;BE LOCKED ON PRIMARY PROCESSOR
008D 1482 : DELPTE1:
50 01 CE 008D 1483 : MNEGL #1,R0 ;INDICATE DECREMENT OF PTWSLELCK ARRAY
FF71 30 0090 1484 : BSBW MMG$MOVPTLOCK ;ONE LESS REASON PT PAGE MUST BE LOCKED
034A 31 0093 1485 : BRW DELPTE ;OK TO DELETE PAGE NOW
0096 1486 :
0096 1487 :
0096 1488 : DELETE PTE TO A SHARED MEMORY GLOBAL SECTION PAGE.
0096 1489 :
0096 1490 : SHMPAGDEL:
0052 8F BB 0096 1491 : PUSHR #^M<R1,R4,R6> ;SAVE REGISTERS
51 01 9A 009A 1492 : MOVZBL #1,R1 ;COUNT TO DECREMENT PTE REFERENCE BY
FF60' 30 009D 1493 : BSBW MMG$FINDGSDPFN ;FIND SHARED MEMORY GSD FOR THIS PFN
0052 8F BA 00A0 1494 : POPR #^M<R1,R4,R6> ;RESTORE REGISTERS
00A4 1495 :
00A4 1496 : AT SOME TIME, A SEND MESSAGE TO THE ERROR LOGGERS COULD BE ADDED HERE.
00A4 1497 :
E6 50 E8 00A4 1498 : BLBS R0,DELPTE1 ;BR ON GSD FOUND, GO DELETE PTE
00A7 1499 : BUG_CHECK NOSHMGSD,FATAL ;FATAL ERROR IF NOT ASSOCIATED WITH GS
00AB 1500 :
00AB 1501 : PURGE WORKING SET SCAN GOT A RESOURCE ERROR
00AB 1502 : O(SP) = SAVED R8
00AB 1503 :
00AB 1504 : WSSCNWAIT:
51 58 D0 00AB 1505 : MOVL R8,R1 ;RESOURCE TO WAIT FOR
58 8E D0 00AE 1506 : MOVL (SP)+,R8 ;RESTORE R8
7E DC 00B1 1507 : MOVPSL -(SP) ;RESOURCE WAIT ROUTINE NEEDS THIS
5E 04 C0 00B3 1508 : ADDL2 #4,SP ;JUNK WORD
50 DD 00B6 1509 : PUSHL R0 ;ACCESS MODE
00B8 1510 : DSBINT #IPL$ SYNCH
015B 31 00BE 1511 : BRW DELPAGRESWAIT
00C1 1512 :
00C1 1513 : PAGE LOCKED IN WORKING SET
00C1 1514 :
00C1 1515 : DELPAGLOCKED:
OC A5 51 B1 00C1 1516 : CMPW R1,PHDSL_WSLOCK(R5) ;IF PERMANENT PAGE, SUCCESSFUL NOP
```



```
58 00090030 22 1F 00C5 1517 BLSSU BRDELPAGSUCCESS ;ALLOWING SWAPPER TO USE DELTVA
50 6E 9A 00C7 1518 MOVZBL (SP),R0 ;RECOVER ACCESS MODE
00CA 1519 ENBINT 4(SP) ;RESTORE IPL TO THAT OF CALLER
DD 00CE 1520 PUSHL R8 ;SAVE R8 AROUND THE FOLLOWING CALL
8F D0 00D0 1521 MOVL #<SS$ WASSET @16 !- ;SET PARAMETER TO UNLOCK THE
FF26' 30 00D7 1522 WSL$M-WSLOCK ! WSL$M_PFNLOCK>,R8 ;WORKING SET LIST ENTRY
0100 8F BA 00DA 1523 BSBW MMG$LCULKPAG ;UNLOCK THE PAGE FROM THE WORKING SET
0B 50 E9 00DE 1524 POPR #^M<R8> ;RESTORE R8
01CC 31 00E1 1525 BLBC R0,BRDELPAGRET ;BRANCH IF ERROR
00E4 1526 BRW DELAGAIN ;GO RESTART DELPAG FROM THE TOP
00E4 1527
00E4 1528 ; LENGTH VIOLATION FOR THIS VIRTUAL ADDRESS
00E4 1529
00E4 1530
00E4 1531 DELPAGLENVIO:
56 D4 00E4 1532 CLRL R6 ;OTHERWISE, FAKE RANGE COMPLETED
0308 31 00E6 1533 BRW DELPAGLAST ;CONTRACT THE REGION IF POSSIBLE
00E9 1534 ;CHECK FOR LAST IN RANGE
00E9 1535 BRDELPAGSUCCESS:
02FA 31 00E9 1536 BRW DELPAGSUCCESS
00EC 1537 BRDELPAGRET:
02FA 31 00EC 1538 BRW DELPAGRET
00EF 1539
00EF 1540 ; R1 = PAGE TYPE AND IS KNOWN NOT TO BE ZERO
00EF 1541
00EF 1542
00EF 1543
00EF 1544 DELPAGNOTPROCES:
00EF 1545 ASSUME PFNS$C_SYSTEM EQ 1
00EF 1546 ASSUME PFNS$C_PPGTBL EQ 4 ;NEGATIVE FIELD VALUE
51 D7 00EF 1547 DECL R1 ;IF 1 OR NEGATIVE
F6 15 00F1 1548 BLEQ BRDELPAGSUCCESS ;THEN, SUCCESSFUL NOP
00F3 1549 ;ALLOWING SWAPPER TO USE DELTVA
00F3 1550
00F3 1551 ; CHECK FOR A RESIDENT GLOBAL SECTION PAGE
00F3 1552
51 0000'DF40 D0 00F3 1553 MOVL @W^PFNS$AL BAK[R0],R1 ;GET BACKING STORE
2E 51 16 E1 00F9 1554 BBC #PTES$V_TYPO,R1,DELGBLRDINPROG ;MUST BE PAGE-FILE
51 51 32 00FD 1555 CVTWL R1,R1 ;ISOLATE GLOBAL SECTION TABLE INDEX
50 00000000'EF D0 0100 1556 MOVL MMG$GL_SYSPHD,R0 ;ADDRESS OF SYSTEM HEADER
50 20 A0 C0 0107 1557 ADDL PHD$PSTBASOFF(R0),R0 ;GET SECTION TABLE BASE
50 6041 DE 010B 1558 MOVAL (R0)[R1],R0 ;GET THE CORRECT SECTION TABLE ENTRY
17 14 A0 0D E1 010F 1559 BBC #SEC$V_RESIDENT,SEC$W_FLAGS(R0),DELGBLRDINPROG
24 BB 0114 1560 PUSHR #^M<R2,R5>
55 00000000'EF D0 0116 1561 MOVL MMG$GL_SYSPHD,R5 ;SYSTEM HEADER
FEE0' 30 011D 1562 BSBW MMG$DECTSECTREF ;ONE LESS REFERENCE
24 BA 0120 1563 POPR #^M<R2,R5>
50 01 CE 0122 1564 MNEGL #1,R0 ;UNLOCK INDICATOR
FEDC 30 0125 1565 BSBW MMG$MOVPTLOCK ;ONE LESS REASON TO LOCK THE PAGE TABLE
02B5 31 0128 1566 BRW DELPTE ;FINISH DELETING THE PAGE TABLE ENTRY
012B 1567
012B 1568 DELGBLRDINPROG:
OB BA 012B 1569 POPR #^M<R0,R1,R3> ;R0=ACCESS MODE, R1=SAVED IPL
012D 1570 ;CLEAN OFF SAVED PSL INTO R3
012D 1571 ENBINT R1 ;BACK TO CALLER'S IPL
58 DD 0130 1572 PUSHL R8 ;SAVE THIS REGISTER
58 D4 0132 1573 CLRL R8 ;SCANNING FOR GLOBAL PAGES
```



```
FEC9' 30 0134 1574 BSBW MMG$PURGWSSCN ;DELETE GLOBAL PAGES IN THE ADDRESS RANGE
      0137 1575 ;FROM THE WORKING SET LIST
      58 D5 0137 1576 TSTL R8 ;CHECK FOR ERROR
      03 13 0139 1577 BEQL 10$
      FF6D 31 013B 1578 BRW WSSCNWAIT ;GO WAIT FOR A RESOURCE
58 8E D0 013E 1579 10$: MOVL (SP)+,R8 ;RESTORE R8
      0141 1580 ;
      0141 1581 ; FALL THROUGH TO DELPAG
      0141 1582 ;
      0141 1583 ;
      0141 1584 .ENABL LSB
      0141 1585
      7E DC 0141 1586 MMG$DELPAG::
      0143 1587 MOVPSL -(SP) ;SAVE PSL
      50 DD 0146 1588 SAVIPL ;PUSH CALLER'S IPL
      FEB5' 30 0148 1589 PUSHL R0 ;SAVE ACCESS MODE
      96 50 E9 014B 1590 DELP1: BSBW MMG$PTEREF ;REFERENCE PTE, RETURN SVAPTE
      014B 1591 ;RETURNS AT IPL=SYNCH
      014B 1592 BLBC R0,DELPAGLENVIO ;BRANCH IF LENGTH VIOLATION
      014E 1593 ;
      014E 1594 ; R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
      014E 1595 ;
      51 63 D0 014E 1596 MOVL (R3),R1 ;FETCH PAGE TABLE ENTRY
      96 13 0151 1597 BEQL BRDELPAJSUCCESS ;BRANCH IF ALREADY DELETED
      50 6E 9A 0153 1598 MOVZBL (SP),R0 ;ACCESS MODE
      02F0 30 0156 1599 BSBW MMG$PAGETYPE ;CHECK ACCESS, RETURN PAGE TYPE
      90 50 E9 0159 1600 BLBC R0,BRDELPAJRET ;BRANCH IF PAGE OWNER VIOLATION
      015C 1601 ;
      015C 1602 ; R1 <0:7> PAGE TYPE TO CASE ON (TYP0*2 + TYP1)
      015C 1603 ; R2 = VA, R3 = SVAPTE, IPL = SYNCH
      015C 1604 ; 0(SP) = ACCESS MODE, 4(SP) = SAVED IPL, 8(SP) = SAVED PSL
      015C 1605 ;
      50 63 16 00 EF 015C 1606 EXTZV #PTESV_PGFLVB,#PTES PGFLVB,(R3),R0 ;PFN, PGFLVB, GPTX, OR SECTION
      0161 1607 CASE TYPE=B, SRC=R1, DISPCIST=<- ;DISPATCH ON THE PTE TYPE
      0161 1608 DELTRANS,- ;PAGE IN TRANSITION (OR DZRO)
      0161 1609 DELPAJFIL,- ;PAGING FILE
      0161 1610 DELGLOBAL,- ;GLOBAL PAGE
      0161 1611 DELSTX- ;SECTION TABLE INDEX
      0161 1612 >
      016D 1613 ;
      016D 1614 ; FALL THROUGH THE CASE STATEMENT FOR VALID PAGE TABLE ENTRY
      016D 1615 ;
      016D 1616 DELVALID:
      016D 1617 ASSUME PTESV_WINDOW LE PTES PGFLVB
      45 50 15 E0 016D 1618 BBS #PTESV_WINDOW,R0,50$ ;BR IF PFNMAP-ED PAGE
      00000000'EF 50 D1 0171 1619 CMPL R0,MMG$GL_MAXMEM ;IS PFN IN SHARED MEMORY?
      3C 1A 0178 1620 BGTRU 50$ ;BR ON YES, GO ALTER SHM GSD REF CNT
      0000'DF40 03 00 EE 017A 1621 EXTIV #PFNSV_PAGTYP,#PFNS PAGTYP, - ;GET PAGE TYPE
      51 0181 1622 @W^PFNSAB TYPE[R0],RT
      0182 1623 ASSUME PFNSC_PROCESS EQ 0
      03 13 0182 1624 BEQL 10$
      FF68 31 0184 1625 BRW DELPAGNOTPROCES ;BRANCH IF NOT PROCESS PAGE
      0187 1626 10$:
      0187 1627 ;
      0187 1628 ; CHECK TO SEE THAT PAGE IS NOT LOCKED IN THE WORKING SET
      0187 1629 ;
      0187 1630 PFN_REFERENCE -
```



```
0187 1631 MOVZWL <@W^PFNSAx WSLX[RO],R1>,- ;INDEX TO WORKING SET LIST ENTRY
0187 1632 LONG_OPCODE=MOVL,-
0187 1633 IMAGE=SYS_NONPAGED
018D 1634 BITL #<WSLSM_WSLOCK ! WSLSM_PFNLOCK>,(R5)[R1] ;LOCKED OR PERMANENT PAGE?
0191 1635 BNEQ 57$ ;BRANCH IF YES
0193 1636 CMPW @W^PFNSAw_REFcnt[RO],#1 ;MUST WAIT IF REFERENCE COUNT IS UP
0199 1637 BGTR 90$ ;INDICATING I/O IN PROGRESS
019B 1638 BBS #PTESV_MODIFY,(R3),40$ ;BRANCH IF PAGE MODIFIED
019F 1639 BBC #PFNSV_MODIFY,@W^PFNSAB STATE[RO],60$ ;BRANCH IF PAGE NOT MODIFIED
01A6 1640 40$: ROTL #<31-PTESV_TYPO>,@W^PFNSAL BAK[RO],R1 ;SECTION BACKING STORE?
01AD 1641 BGEQ 60$ ;BRANCH IF NOT
01AF 1642 BBS #<31-<PTESV_TYPO-PTESV_CHKPT>>,R1,60$ ;IS IT MARKED CHECKPOINTED
01B3 1643 BRW DELPAG_WRTBAK ;MUST WRITE THIS PAGE BACK
01B6 1644 50$: BRW PFNMAPDEL ;BR TO DELETE PFNMAP-ED PAGE
01B9 1645 57$: BRW DELPAGLOCKED ;BR TO DELETE LOCKED PAGE
01BC 1646 DELPAG_WRTDONE:
01BC 1647 60$: BICB3 #PFNSM_MODIFY,@W^PFNSAB STATE[RO],R1 ;FETCH ALL BUT MODIFY BIT
01C4 1648 BISB3 #PFNSM_DELCON,R1,@W^PFNSAB STATE[RO] ;SET DELCON, CLEAR MODIFY
01CB 1649 BICB #<PTESM_VALID ! PTESM_MODIFY>@-24,3(R3) ;MAKE PTE INTO TRANSITION FO
01D0 1650 INVALID R2 ;AND INVALIDATE THE TRANSLATION BUFFER
01D3 1651 PUSHL R2 ;SAVE VA
01D5 1652
01D5 1653 MOVZWL <@W^PFNSAx WSLX[RO],R1>,- ;GET WORKING SET LIST INDEX
01D5 1654 LONG_OPCODE=MOVL,-
01D5 1655 IMAGE=SYS_NONPAGED
01DB 1656 XORL3 R2,(R5)[R1],R3 ;HIGH VA BITS MUST AGREE
01E0 1657 ASHL #-VASS_BYTE,R3,R3 ;SHIFT OUT THE ONES THAT MAY DIFFER
01E5 1658 BNEQ WSLXVANOMAT ;AND BRANCH IF HIGH BITS DISAGREE
01E7 1659 BSBW MMGSDELWSLEPPG ;DELETE WORKING SET LIST ENTRY
01EA 1660 ;FOR PRCESS PAGE
01EA 1661 :
01EA 1662 : IF THE PAGE WAS QUEUED FOR WRITING, THEN THE REFERENCE COUNT WILL BE UP
01EA 1663 :
01EA 1664 DECREF GTR=80$ ;ONE LESS REF WITH WSLE RELEASED
01F6 1665 BSBW MMGSRELPFN ;WILL RESULT IN PTE CONTAINING ITS
01F9 1666 ;BACKING STORE FORMAT
01F9 1667 BRW DELAGAIN1 ;GO DELETE THE PTE
01FC 1668 :
01FC 1669 : WORKING SET LIST ENTRY AND VIRTUAL ADDRESS DISAGREE
01FC 1670 :
01FC 1671 WSLXVANOMAT:
01FC 1672 BUG_CHECK WSLXVANMAT,FATAL
0200 1673 :
0200 1674 : I/O IN PROGRESS ON THIS PAGE, MUST WAIT
0200 1675 :
0200 1676 80$: POPR #^M<R2> ;RESTORE SAVED VA
0202 1677 90$: BRB DELPAGIOWAIT ;WAIT FOR I/O COMPLETION
0204 1678
0204 1679 .DSABL LSB
0204 1680 :
0204 1681 : WAIT FOR I/O TO COMPLETE OR RESOURCE TO BECOME AVAILABLE
0204 1682 : 0(SP) = SAVED ACCESS MODE, TO BE RESTORED TO R0
0204 1683 : 8(SP) = PSL AT ENTRY TO DELPAG
0204 1684 : R2, R4, R5, R6, R7, R8 AS THEY WERE ON ENTRY TO DELPAG
0204 1685 : R1 = RESOURCE TO WAIT FOR
0204 1686 :
0204 1687 DELPAGASTWAIT:
```



```

      24 FC AD 0A E0 0204 1688      BBS      #MMGSV NOWAIT IPL0,-      ;RETURN SSS_ABORT
      51 01 9A 0206 1689      MMGSL MAXACMODE(FP),DELPAGIOWAIT
      52 DD 0209 1690      MOVZBL #RSNS_ASTWAIT,R1      ;WAIT ON AST'S
      FDEF' 30 020C 1691      PUSHL R2      ;SAVE VIRTUAL ADDRESS
53  OC AE 001F0000 8F CB 020E 1692      BSBW MMG$RESRCWAIT      ;SET UP TO WAIT FOR THE RESOURCE
      36 11 0211 1693      BICL3 #PSLSM IPL,12(SP),R3      ;THIS WILL BE IPL 0 WAIT
      021A 1694      BRB DELPAGIOWAIT4
      021C 1695      ;
      021C 1696      DELPAGRESWAIT:
      52 DD 021C 1697      PUSHL R2      ;SAVE VIRTUAL ADDRESS
      FDDF' 30 021E 1698      BSBW MMG$RESRCWAIT      ;SET UP TO WAIT FOR THE RESOURCE
      04 BA 0221 1699      POPR #^M<R2>      ;RESTORE SAVED VIRTUAL ADDRESS
      01 BA 0223 1700      POPR #^M<R0>      ;R0 = SAVED ACCESS MODE
      6E FF18 CF DE 0225 1701      MOVAL W^MMG$DELPAG,(SP)      ;STORE PC TO RESTART AT
      FDD3' 31 022A 1702      BRW MMG$SVPCTX      ;AND EXIT TO SCHEDULER
      022D 1703
      022D 1704      ;
      022D 1705      ; R0 = PAGE FRAME NUMBER
      022D 1706      ;
      022D 1707      DELPAGIOWAIT:
53  08 AE 001F0000 8F CB 022D 1708      BICL3 #PSLSM IPL,8(SP),R3      ;THIS WILL BE IPL 0 WAIT
      0A E1 0236 1709      BBC      #MMGSV NOWAIT IPL0,-      ;OK TO WAIT AT IPL 0
      06 FC AD 0238 1710      MMGSL MAXACMODE(FP),DELPAGIOWAIT2
      50 2C 3C 023B 1711      MOVZWL #SS$ ABORT,R0
      01A8 31 023E 1712      BRW DELPAGRET
      0241 1713      ;
      0241 1714      DELPAGIOWAIT2:
      0000'DF40 40 8F 88 0241 1715      BISB #PFNSM_RPTEVT,@W^PFNSAB_TYPE[R0] ;WAKE THIS PROCESS ON I/O COMPLETE
      52 DD 0248 1716      PUSHL R2      ;DELCONPFN WILL CLEAR THIS BIT
      50 0000'CF 7E 024A 1717      MOVAQ W^SCH$GQ PFWQ,R0      ;SAVE VIRTUAL ADDRESS
      FDAE' 30 024F 1718      BSBW MMG$PGFLTWAIT      ;PAGE FAULT WAIT QUEUE HEAD
      04 BA 0252 1719      POPR #^M<R2>      ;SET UP TO WAIT ON PAGE FAULT WAIT QUEUE
      53 DD 0252 1720      DELPAGIOWAIT4:
      00000000'EF 16 0252 1721      POPR #^M<R2>      ;RESTORE SAVED VIRTUAL ADDRESS
      FEE6 31 0254 1722      PUSHL R3      ;WAIT AT SPECIFIED IPL - NOTE THAT
      0256 1723      ;AST'S MAY OCCUR AT ANY TIME IF IPL 0
      0256 1724      JSB MMG$SVPCTX      ;AND CALL SCHEDULER
      025C 1725      SETIPL #IPL$ ASTDEL      ;RAISE BACK TO ASTDEL
      025F 1726      BRW DELP1      ;TRY TO DELETE THE PAGE AGAIN
      0262 1727      ;
      53 08 AE D0 0262 1728      DELPAGIOWAIT3:
      D9 11 0262 1729      MOVL 8(SP),R3      ;WAIT AT IPL OF CALLER
      0266 1730      BRB DELPAGIOWAIT2      ;GO WAIT
      0268 1731      ;
      0268 1732      ; TRANSITION PAGE ON FREE, MODIFIED OR BAD PAGE LIST
      0268 1733      ; R0 = PFN, R1 = LIST ID
      0268 1734      ;
      0268 1735      MFY_OR_FRELIST:
      52 DD 0268 1736      PUSHL R2      ;SAVE VA
      51 D0 026A 1737      MOVL R1,R2      ;GET LIST ID
      17 13 026D 1738      BEQL 20$      ;BRANCH IF FREE LIST
      OD 52 E9 026F 1739      BLBC R2,10$      ;BRANCH IF BAD PAGE LIST
51  0000'DF40 09 9C 0272 1740      ROTL #<31-PTESV_TYPO>,@W^PFNSAB_BAK[R0],R1 ;TYPO BIT TO SIGN BIT
      04 18 0279 1741      BGEQ 10$      ;BRANCH IF NOT SECTION PAGE (TYPO=0)
      04 BA 027B 1742      POPR #^M<R2>      ;RECOVER SAVED VA
      2B 11 027D 1743      BRB DELFAULT      ;AND FAULT THE SECTION PAGE
      0000'DF40 80 8F 8A 027F 1744 10$:      BICB #PFNSM_MODIFY,@W^PFNSAB_STATE[R0] ;OTHERWISE SKIP THE WRITE
```



```
FD77' 30 0286 1745 20$: BSBW MMG$DELPFNLST ;REMOVE PAGE FROM LIST AND DELETE CONTENTS
      04 BA 0289 1746 DELAGAIN1:
      23 11 0289 1747 POPR #^M<R2> ;RECOVER SAVED VA
      028B 1748 BRB DELAGAIN
      028D 1749 ;
      028D 1750 ; THIS IS A PAGE IN TRANSITION OR A DEMAND ZERO PAGE
      028D 1751 ; R0 = PFN, R2 = VA, R3 = SVAPTE
      028D 1752 ;
      028D 1753 DELTRANS:
      50 D5 028D 1754 TSTL R0 ;ZERO PFN INDICATES DEMAND ZERO PAGE
      39 13 028F 1755 BEQL BRBINCPGFLCNT ;BRANCH IF IT IS
      0291 1756
      0291 1757 ASSUME PFN$C_FREPAGLST EQ 0
      0291 1758 ASSUME PFN$C_MFY PAGLST EQ 1
      0291 1759 ASSUME PFN$C_BADPAGLST EQ 2
      0291 1760 ASSUME PFN$C_RELPEND EQ 3
      0291 1761 ASSUME PFN$C_RDERR EQ 4 :-4
      0291 1762 ASSUME PFN$C_WRTINPROG EQ 5 :-3
      0291 1763 ASSUME PFN$C_RDINPROG EQ 6 :-2
      0291 1764 ASSUME PFN$C_ACTIVE EQ 7 :-1
      0291 1765
      51 03 00 EE 0291 1766 EXTV #PFN$V LOC,#PFN$S LOC,- ;GET PAGE'S LOCATION
      0000'DF40 0294 1767 @W^PFN$AB_STATE[R0],R1
      0299 1768 CASE R1,<-
      0299 1769 DELPAGIOWAIT3,- :-3 = 5 = WRITE IN PROGRESS
      0299 1770 DELFAULT,- :-2 = 6 = READ IN PROGRESS
      0299 1771 DELVALID,- :-1 = 7 = ACTIVE
      0299 1772 MFY_OR_FRELST,- ;FREE PAGE LIST
      0299 1773 MFY_OR_FRELST,- ;MODIFIED PAGE LIST
      0299 1774 MFY_OR_FRELST,- ;BAD PAGE LIST
      0299 1775 >,TYPE=B,LIMIT=#-3
      02AA 1776 DELFAULT:
      02AA 1777 ENBINT 4(SP) ;OTHERWISE RESTORE IPL
      02AE 1778 ;
      02AE 1779 ; FAULT THE PAGE OUT OF THE TRANSITION STATE
      02AE 1780 ;
      62 95 02AE 1781 TSTB (R2) ;REFERENCE THE PAGE
      02B0 1782 DELAGAIN:
      01 BA 02B0 1783 POPR #^M<R0> ;RESTORE R0 = ACCESS MODE
      02B2 1784 ENBINT ;POP AND RESTORE SAVED IPL
      5E 04 C0 02B5 1785 ADDL #4,SP ;CLEAN OFF SAVED PSL
      FE86 31 02B8 1786 BRW MMG$DELPAG ;RESTART THE ROUTINE FROM THE TOP
      02BB 1787 ;
      02BB 1788 ; THIS IS A PAGE FILE PTE
      02BB 1789 ; R0 = PAGE FILE VBN
      02BB 1790 ;
      02BB 1791 DELPAGFIL:
      50 D5 02BB 1792 TSTL R0 ;NULL PAGE FILE VBN?
      0B 13 02BD 1793 BEQL BRBINCPGFLCNT ;BRANCH IF YES
      0C BB 02BF 1794 PUSHR #^M<R2,R3> ;SAVE VA, SVAPTE
      53 1F A5 9A 02C1 1795 MOVZBL PHD$B_PAGFIL(R5),R3 ;PAGE FILE INDEX IN R1
      FD38' 30 02C5 1796 BSBW MMG$D$LC_PAGFIL ;DEALLOCATE THE PAGING FILE VBN
      0C BA 02C8 1797 POPR #^M<R2,R3> ;RESTORE R2 = VA, R3 = SVAPTE
      02CA 1798 BRBINCPGFLCNT:
      010B 31 02CA 1799 BRW INCPGFLCNT ;GO DELETE THE PTE
      02CD 1800 ;
      02CD 1801 ; THIS IS A GLOBAL PTE
```



```
51 0000'DF40 7B800000 8F CB 02CD 1802 ; RO = GLOBAL PAGE TABLE INDEX
                                02CD 1803 ;
                                02CD 1804 DELGLOBAL:
                                02CD 1805 .ENABL LSB
                                02CD 1806 BICL3 #^C<PTESM_VALID ! - ;VALID BIT
                                02D8 1807 PTESM_TYPT ! PTESM_TYPO ! - ;PAGE TYPE BITS
                                02D8 1808 PTESM_PGFLVB>, @W^MMG$GL GPTBASE[R0], R1 ; ETC, FROM MASTER PTE
50 51 15 00 EF 02D8 1809 EXTZV #PTESV_PFN, #PTESV_PFN, RT, RO ; PFN FROM VALID OR TRANSITION PTE
                                10 13 02DD 1810 BEQL 12$ ; PFN OF 0 SHOULD ONLY BE DZERO
                                22 51 1F E0 02DF 1811 BBS #PTESV_VALID, R1, 20$ ; BRANCH IF MASTER PTE IS VALID
                                3E 51 1A E0 02E3 1812 BBS #PTESV_TYPT, R1, 30$ ; BRANCH IF NOT TRANSITION PAGE
                                OA 51 16 E1 02E7 1813 BBC #PTESV_TYPO, R1, 15$ ; BRANCH IF TRANSITION PTE
                                02EB 1814 10$: BUG_CHECK IVGBCTYP, FATAL ; INVALID GLOBAL MASTER PTE TYPE
                                02EF 1815
                                51 D5 02EF 1816 12$: TSTL R1 ; MAKE SURE IT IS DZERO
                                71 13 02F1 1817 BEQL 70$ ; GO DELETE IT
                                F6 11 02F3 1818 BRB 10$
                                02F5 1819 ;
                                02F5 1820 ; TRANSITION PAGE TABLE ENTRY
                                02F5 1821 ;
51 03 00 EE 02F5 1822 15$: EXTV #PFNSV_LOC, #PFNS$ LOC, - ; IF PAGE IS READ IN PROGRESS
                                0000'DF40 02F8 1823 @W^PFNS$AB_STATE[R0], R1 ; READ ERROR, OR ACTIVE BUT NOT VALID
                                02FD 1824 ; THEN MUST SEE IF THERE IS A WSLE
                                51 FD 06 18 02FD 1825 BGEQ 20$ ; BRANCH IF ON PFN LIST OR RELPEND
                                8F 91 02FF 1826 CMPB #<PFNSC_WRTINPROG ! <<-1> @PFNS$ LOC>>, R1 ; IF NOT WRITE IN PROGRESS
                                48 12 0303 1827 BNEQ 40$ ; THEN READ IN PROG, READ ERR, ACTIVE
                                0305 1828 ;
                                0305 1829 ; SEE IF I/O IS IN PROGRESS ON THIS PAGE
                                0305 1830 ;
                                51 D4 0305 1831 20$: CLRL R1 ; COMPARE REFCNT AGAINST 0
                                0307 1832 PFN_REFERENCE -
                                0307 1833 TSTW <@W^PFNS$X_SHRCNT[R0]>, - ; IF SHRCNT IS 0
                                0307 1834 LONG_OPCODE=TSTL, -
                                0307 1835 IMAGE=SYS_NONPAGED
                                02 13 030C 1836 BEQL 22$
                                51 D6 030E 1837 INCL R1 ; OTHERWISE COMPARE REFCNT AGAINST 1
51 0000'DF40 B1 0310 1838 22$: CMPW @W^PFNS$AW_REFCNT[R0], R1 ; I/O IN PROGRESS ON THIS PAGE?
                                07 15 0316 1839 BLEQ 26$ ; BRANCH IF NOT
                                3E A4 40 A4 B1 0318 1840 CMPW PCBSW_DIO_LM(R4), PCBSW_DIO_CNT(R4) ; ANY DIRECT I/O'S FOR THIS PROCESS?
                                38 1A 031D 1841 BGTRU 50$ ; BRANCH IF YES
51 0000'DF40 D0 031F 1842 26$: MOVL @W^PFNS$AL_BAK[R0], R1 ; BACKING STORE ADDRESS
                                34 51 16 E1 0325 1843 30$: BBC #PTESV_TYPO, R1, 60$ ; BRANCH IF NOT SECTION TABLE
                                27 51 11 E0 0329 1844 BBS #PTESV_DZRO, R1, DELFAULT1 ; FAULT IF DEMAND ZERO
                                08 51 10 E1 032D 1845 BBC #PTESV_CRF, R1, 35$ ; BRANCH IF NOT COPY ON REFERENCE SECTION
50 0080 C4 D0 0331 1846 MOVL PCBSL_JIB(R4), R0 ; GET JIB ADDRESS
                                3C A0 D6 0336 1847 INCL JIB$ PGFLCNT(R0) ; ONE LESS PAGE FILE PAGE CHARGED
                                51 51 32 0339 1848 35$: CVTWL R1, R1 ; ISOLATE GLOBAL SECTION TABLE INDEX
                                BB 033C 1849 PUSHR #^M<R2, R5> ; SAVE THESE
55 00000000'FF DE 033E 1850 MOVAL @L^MMG$GL_SYSPHD, R5 ; ADDRESS OF SYSTEM HEADER
                                FCB8' 30 0345 1851 BSBW MMG$DECSECTREF ; COUNT ONE LESS SECTION REFERENCE
                                24 BA 0348 1852 POPR #^M<R2, R5> ; RESTORE SAVED REGISTERS
                                0093 31 034A 1853 BRW DELPTE ; FINISH DELETING THE PAGE TABLE ENTRY
                                B4 6E 09 E2 034D 1854 40$: BBSS #MMG$V_DELGBLDON, (SP), 20$ ; IF ALREADY DELETED THESE GLOBAL
                                FDD7 31 0351 1855 BRW DELGBLRDINPROG ; WSLE'S, THEN DON'T DO IT AGAIN
                                0354 1856 ; GO DO THE DELGBL SCAN
                                0354 1857
                                0354 1858 DELFAULT1:
```



```
FF53 31 0354 1859 BRW DELFAULT ;BR TO FAULT PAGE
      0357 1860
      0357 1861 ;
      0357 1862 ; THIS GLOBAL PAGE HAS I/O OUTSTANDING AND THIS PROCESS HAS SOME
      0357 1863 ; DIRECT I/O'S IN PROGRESS. JUST IN CASE HE IS RESPONSIBLE FOR THE
      0357 1864 ; I/O REQUEST ON THIS PAGE, MAKE HIM WAIT FOR ALL HIS DIRECT I/O
      0357 1865 ; REQUESTS TO COMPLETE.
      0357 1866 ;
51 01 3C 0357 1867 50$: MOVZWL #RSNS$ASTWAIT,R1 ;WAIT FOR AN AST TO OCCUR
FEBF 31 035A 1868 BRW DELPAGRESWAIT
      035D 1869
      035D 1870 ;
      035D 1871 ; HANDLE THE POSSIBILITY OF A GLOBAL SECTION WITH PAGE FILE BACKING STORE
      035D 1872 ;
      035D 1873 60$:
00 51 02 16 ED 035D 1874 CMPZV #PTESV_TYPO,#2,R1,#0 ;CHECK FOR PAGE FILE
      87 12 0362 1875 BNEQ 10$
      02 10 0364 1876 70$: BSB FNDGBLSECTBL ;FIND SECTION REF COUNT & DECREMENT IT
      78 11 0366 1877 BRB DELPTE
      0368 1878
      0368 1879 .DSABL LSB
      0368 1880 ;
      0368 1881 ; FIND THE GLOBAL SECTION TABLE FOR A GIVEN GPTX
      0368 1882 ; DECREMENT THE REFERENCE COUNT
      0368 1883 ;
      0368 1884 FNDGBLSECTBL:
55 00E4 8F BB 0368 1885 PUSHR #^M<R2,R5,R6,R7>
56 00000000'FF DE 036C 1886 MOVAL @L^MMG$GL_SYSPHD,R5 ;ADDRESS OF SYSTEM HEADER
50 63 16 00 EF 0373 1887 EXTZV #PTESV_PGFLVB,#PTESV_PGFLVB,(R3),R6 ;GET THE GPT INDEX
51 55 20 A5 C1 0378 1888 ADDL3 PHD$L_PSTBASOFF(R5),R5,R0 ;BASE OF SECTION TABLE
      00000000'EF D0 037D 1889 MOVL MMG$GC_GBLSECFND,R1 ;DO WE HAVE A PREVIOUS MATCH TO START WITH
      04 12 0384 1890 BNEQ 10$ ;YES
      51 24 A5 32 0386 1891 5$: CVTWL PHD$W_PSTLAST(R5),R1 ;INDEX TO LAST SECTION
      52 6041 DE 038A 1892 10$: MOVAL (R0)[R1],R2 ;SECTION ADDRESS
      18 A2 D5 038E 1893 TSTL SECL_REFCNT(R2) ;IGNORE IF NEGATIVE REF COUNT
      19 15 0391 1894 BLEQ 20$
      16 00 EF 0393 1895 EXTZV #SECSV_VPX,#SECSV_VPX,- ;BASE FOR THIS SECTION
57 08 A2 0396 1896 SECL_VPXPC(R2),R7
      57 56 D1 0399 1897 CMPL R6,R7 ;WRONG ONE
      0E 19 039C 1898 BLSS 20$
57 1C A2 C0 039E 1899 ADDL2 SECL_PAGCNT(R2),R7
      57 56 D1 03A2 1900 CMPL R6,R7 ;WRONG ONE
      05 18 03A5 1901 BGEQ 20$
      FC56' 30 03A7 1902 BSBW MMG$DECSECTREF ;GO DEC THE REF COUNT
      15 11 03AA 1903 BRB 30$
00000000'EF D5 03AC 1904 20$: TSTL MMG$GL_GBLSECFND ;IS THIS THE FIRST LOOP
      08 13 03B2 1905 BEQL 25$ ;NO
00000000'EF D4 03B4 1906 CLRL MMG$GL_GBLSECFND
      CA 11 03BA 1907 BRB 5$
      51 08 C0 03BC 1908 25$: ADDL2 #SECSV_LENGTH-2,R1 ;NEXT SECTION
      C9 19 03BF 1909 BLSS 10$
      00E4 8F BA 03C1 1910 30$: POPR #^M<R2,R5,R6,R7>
      05 03C5 1911 RSB
      03C6 1912 ;
      03C6 1913 ; THIS IS A PROCESS SECTION PAGE
      03C6 1914 ; R0 = SECTION INDEX AND FLAGS
      03C6 1915 ;
```



```

      8A 50 11 E0 03C6 1916 DELSTX:
      51 50 32 03C6 1917 BBS #PTESV_DZRO,R0,DELFAULT1 ;PROCESS SECTION PAGE
      52 DD 03CA 1918 CVTTL R0,R1 ;FAULT IF DEMAND ZERO
      FC2E' 30 03CD 1919 PUSHL R2 ;SECTION INDEX
      04 BA 03CF 1920 BSBW MMGSDECSECREP ;SAVE VA
      08 50 10 E1 03D2 1921 POPR #^M<R2> ;DECREASE SECTION REFERENCE COUNT
      50 0080 C4 D0 03D4 1922 BBC #PTESV_CRF,R0,DELPTE ;RECOVER SAVED VA
      3C A0 D6 03D8 1923 INCPGFLCNT: ;BRANCH IF NOT CRF SECTION
      50 0080 C4 D0 03D8 1924 MOVL PCBSL_JIB(R4),R0 ;GET JIB ADDRESS
      3C A0 D6 03DD 1925 INCL JIBSL_PGFLCNT(R0) ;ONE LESS PAGE FILE PAGE CHARGED
      03E0 1926 ;
      03E0 1927 ; FALL THROUGH TO DELPTE
      03E0 1928 ;
      03E0 1929 DELPTE:
      63 D4 03E0 1930 CLRL (R3) ;JUST ZERO THE PTE
      56 D5 03E2 1931 TSTL R6 ;LAST PAGE OF THE RANGE?
      0B 13 03E4 1932 BEQL DELPAGLAST ;YES, GO CONTRACT THE REGION
      03E6 1933
      50 01 3C 03E6 1934 DELPAGSUCCESS:
      03E6 1935 MOVZWL #SS$_NORMAL,R0 ;SUCCESS RETURN
      03E9 1936 ;
      03E9 1937 ; 0(SP) = ACCESS MODE, 4(SP) = IPL TO RESTORE, 8(SP) = SAVED PSL
      03E9 1938 ;
      03E9 1939 DELPAGRET:
      5E 0C C0 03E9 1940 ENBINT 4(SP) ;RESTORE SAVED IPL
      05 05 03ED 1941 ADDL #12,SP ;ADJUST THE STACK
      03F0 1942 RSB ;AND RETURN
      03F1 1943 ;
      03F1 1944 ; LAST PAGE OF THE DELETE RANGE
      03F1 1945 ;
      03F1 1946 DELPAGLAST:
      0000056D'EF 16 03F1 1947 SETIPL #IPL$_ASTDEL ;NO NEED FOR SYNCH HERE
      EA 11 03F4 1948 JSB MMGS$CONTRACT ;CONTRACT REGION IF POSSIBLE
      03FA 1949 BRB DELPAGSUCCESS ;AND EXIT SUCCESSFULLY
      03FC 1950 ;
      03FC 1951 ; MUST WRITE SECTION PAGE BACK TO DISK BEFORE DELETE
      03FC 1952 ; 0(SP) = SAVED ACCESS MODE
      03FC 1953 ; 8(SP) = SAVED PSL AT ENTRY TO DELPAG
      03FC 1954 ;
      03FC 1955 DELPAG_WRTBAK:
      0D BB 03FC 1956 PUSHR #^M<R0,R2,R3> ;SAVE SOME REGISTERS
      3E A4 B5 03FE 1957 TSTW PCBSW_DIOCNT(R4) ;EXHAUSTED DIRECT I/O QUOTA?
      3E 13 0401 1958 BEQL 40$ ;BRANCH IF YES, WAIT FOR SOME TO COMPLETE
      52 0000'DF 0F 0403 1959 REMQUE @W^IOC$GL_IRPFL,R2 ;GET AN I/O PACKET FROM THE SIDE LIST
      0D 1C 0408 1960 BVC 20$ ;BRANCH IF GOT ONE
      51 C4 8F 9A 040A 1961 MOVZBL #IRP$C_LENGTH,R1 ;SIZE OF PACKET TO ALLOCATE
      FB'EF' 30 040E 1962 BSBW EXESALONONPAGED ;ALLOCATE IT FROM NON-PAGED POOL
      51 03 9A 0411 1963 MOVZBL #RSN$_NPDYNMEM,R1 ;IN ANTICIPATION OF ALLOCATION FAILURE
      2D 50 E9 0414 1964 BLBC R0,60$ ;BRANCH IF MUST WAIT FOR A PACKET
      51 52 D0 0417 1965 20$: MOVL R2,R1 ;I/O PACKET ADDRESS TO R1
      0D BA 041A 1966 POPR #^M<R0,R2,R3> ;R0=PFN, R2=VA, R3=SVAPTE
      0B A1 C4 8F 9B 041C 1967 MOVZBW #IRP$C_LENGTH,IRP$W_SIZE(R1) ;SET SIZE FIELD
      0421 1968 ;TYPE FIELD FILLED IN BY WRTPG$BAK
      0B A1 6E 90 0421 1969 MOVB (SP),IRP$B_RMOD(R1) ;REQUESTING MODE FROM ACCESS MODE PARAM
      48 A1 52 D0 0425 1970 MOVL R2,IRP$S_SEGVBN(R1) ;SAVE STARTING VA FOR SCAN
      0D BB 0429 1971 PUSHR #^M<R0,R2,R3> ;PRESERVE THESE ACROSS THE CALL
      52 0000'DF40 D0 042B 1972 MOVL @W^PFN$AL_BAK[R0],R2 ;BACKING STORE ADDRESS IN R2
```



```
03 00 05 FBCC' 30 0431 1973 BSBW MMGSWRTPGSBAK ;WRITE A CLUSTER OF PAGES BACK
    00 05 OD BA 0434 1974 POPR #^M<R0,R2,R3> ;RESTORE SAVED REGISTERS
    0000'DF40 FO 0436 1975 INSV #PFNSC_WRTINPROG,#PFNSV_LOC,#PFNSS_LOC,-
    FD7B 31 043A 1976 @W^PFNSAB_STATE[R0] ;FORCE THIS PAGE TO WRITE IN PROGRESS
    043E 1977 BRW DELPAG_WRTDONE
    0441 1978 :
    0441 1979 : DIRECT I/O QUOTA EXCEEDED, MUST WAIT FOR A DIRECT I/O TO COMPLETE
    0441 1980 :
    51 01 9A 0441 1981 40$: MOVZBL #RSNS_ASTWAIT,R1 ;WAIT FOR AN AST TO OCCUR
    0444 1982 :
    0444 1983 : UNABLE TO WRITE SECTION PAGE BACK, MUST WAIT FOR RESOURCE IN R1
    0444 1984 :
    OD BA 0444 1985 60$: POPR #^M<R0,R2,R3> ;R0=PFN, R2=VA, R3=SVAPTE
    FDD3 31 0446 1986 BRW DELPAGRESWAIT ;GO WAIT FOR THE RESOURCE
```

```
0449 1988 .SBTTL CONTRACT - ADJUST THE LENGTH OF THE SPECIFIED REGION
0449 1989
0449 1990 :
0449 1991 : INPUTS:
0449 1992 :
0449 1993 : R2 = VIRTUAL ADDRESS (USED ONLY TO SPECIFY P0/P1 SPACE)
0449 1994 : R4 = PROCESS CONTROL BLOCK ADDRESS
0449 1995 : R5 = PROCESS HEADER ADDRESS (P1 SPACE REQUIRED)
0449 1996 : IPL = ASTDEL
0449 1997 :
0449 1998 : OUTPUTS:
0449 1999 :
0449 2000 : R2 PRESERVED
0449 2001 :
0449 2002 :
0449 2003 : *****
0449 2004 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0449 2005 :
0449 2006 :
0000056D 2007 : .PSECT Y$EXEPAGED
056D 2008 :
056D 2009 : *****
056D 2010 :
056D 2011 : MMG$CONTRACT:
52 DD 056D 2012 : PUSH L R2 ;SAVE VIRTUAL ADDRESS
FABE' 30 056F 2013 : BSBW MMG$DALCSTXSCN ;SEE IF ANY SECTIONS TO DEALLOCATE
04 BA 0572 2014 : POPR #^M<R2> ;RESTORE VIRTUAL ADDRESS
50 01 CE 0574 2015 : MNEGL #1,R0 ;INITIAL COUNT OF PAGES TO REDUCE THE REGION
14 52 1E E0 0577 2016 : BBS #VASV_P1,R2,40$ ;BRANCH IF P1 SPACE
057B 2017 :
057B 2018 : P0 SPACE
057B 2019 :
53 16 09 EE 057B 2020 : EXTV #VASV_VPN,#VASS_VPN+1,- ;GET VIRTUAL PAGE NUMBER
53 53 28 A5 057E 2021 : PHD$F-FREP1VA(R5),R3 ;OF FIRST FREE PAGE IN P0 SPACE
00000000'EF C0 0581 2022 : SGNSGC_P1LWCNT,R3 ;AND FORM PAGE TABLE INDEX
50 DD 0588 2023 : R0 ;-1 FOR GOING BACKWARDS THROUGH PTE'S
51 09 DB 058A 2024 : MFPR #PRS_POLR,R1 ;NUMBER OF PAGES IN P0 SPACE
21 11 058D 2025 : BRB 70$ ;START SCAN OF ZERO PTE'S
058F 2026 :
058F 2027 : P1 SPACE
058F 2028 :
53 16 09 EE 058F 2029 40$: EXTV #VASV_VPN,#VASS_VPN+1,- ;VIRTUAL PAGE NO.
53 53 30 A5 0592 2030 : PHD$F-FREP1VA(R5),R3 ;OF FIRST FREE PAGE IN P1 SPACE
00000000'EF C0 0595 2031 : ADDL SGNSGC_P1LWCNT,R3 ;FORM PT INDEX
01 DD 059C 2032 : #1 ;+1 FOR GOING FORWARD THROUGH PTE'S
51 00200000 8F 00D4 C5 C3 059E 2033 : SUBL3 PHD$L_P1LR(R5),#1@21,R1 ;NO. OF PAGES IN P1 SPACE
06 11 05A8 2034 : BRB 70$ ;START SCANNING FOR ZERO PTE'S
05AA 2035 :
05AA 2036 : 0(SP) = +1 IF GOING FORWARD IN PTE'S (P1 SPACE)
05AA 2037 : = -1 IF GOING BACKWARDS IN PTE'S (P0 SPACE)
05AA 2038 : R0 = -1 WITH LOOP ENTERED AT 70$
05AA 2039 : R1 = MAX NUMBER OF PAGE TABLE ENTRIES TO SCAN (POSSIBLY 0)
05AA 2040 : R3 = PTE INDEX FOR THE FIRST NON-EXISTING PAGE IN THE REGION
05AA 2041 :
6C B443 D5 05AA 2042 60$: TSTL @PCB$L_PHD(R4)[R3] ;DELETED PAGE TABLE ENTRY?
07 12 05AE 2043 : BNEQ 80$ ;BRANCH IF NOT
53 6E C0 05B0 2044 70$: ADDL (SP),R3 ;NEXT PAGE TABLE INDEX
```



```
F3 50 51 F2 05B3 2045 AOBLS R1,R0,60$ ;TEST NEXT PTE
05B7 2046 :
05B7 2047 : R0 = NUMBER OF PAGES TO CONTRACT THE REGION BY, POSSIBLY 0
05B7 2048 :
2C A5 50 C0 05B7 2049 80$: ADDL R0,PHD$L_FREPTCNT(R5) ;THAT MANY MORE FREE PAGE TABLE ENTRIES
51 50 09 9C 05B8 2050 ROTL #9,R0,R1 ;NO. OF BYTES TO CONTRACT REGION BY
8E D5 05BF 2051 TSTL (SP)+ ;CLEAN OFF PTE SCAN DIRECTION
12 14 05C1 2052 BGTR 90$ ;BRANCH IF P1 SPACE
05C3 2053 :
05C3 2054 : P0 SPACE
05C3 2055 :
00CC C5 50 C2 05C3 2056 SUBL R0,PHD$L_POLRASTL(R5) ;ADJUST NO. OF PAGES
28 A5 51 C2 05C8 2057 SUBL R1,PHD$L_FREPOVA(R5) ;ADJUST FIRST FREE P0 SPACE VIRT ADR
1B 19 05CC 2058 BLSS 120$ ;ERROR IF NOT STILL P0 SPACE
09 00CC C5 DA 05CE 2059 LDPOLR PHD$L_POLRASTL(R5) ;LOAD HARDWARE P0 SPACE LENGTH REGISTER
10 11 05CE 2060 MTPR PHD$L_POLRASTL(R5),S^#PR$_POLR
05D3 2061 BRB 100$
05D5 2062 :
05D5 2063 : P1 SPACE
05D5 2064 90$: ADDL R0,PHD$L_P1LR(R5) ;UPDATE P1 LENGTH
00D4 C5 50 C0 05D5 2065 ADDL R1,PHD$L_FREP1VA(R5) ;AND FIRST FREE P1 VIRT ADR
30 A5 51 C0 05DA 2066 BLSS 120$ ;ERROR IF NOT STILL P1 SPACE
09 19 05DE 2067 LDP1LR PHD$L_P1LR(R5) ;UPDATE HARDWARE P1 LENGTH REGISTER
0B 00D4 C5 DA 05E0 2068 MTPR PHD$L_P1LR(R5),S^#PR$_P1LR
FF61 30 05E5 2069 100$: BSBW UPDATPTCNT ;UPDATE MAX PAGE TABLE COUNT
05 05E8 2070 RSB
05E9 2071 :
05E9 2072 : FREE P0/P1 VIRTUAL ADDRESS IS FOULED UP
05E9 2073 120$: BUG_CHECK CONTRACT,FATAL ;BAD FREPOVA OR FREP1VA
```

```
05ED 2075      .SBTTL PAGETYPE - CALCULATE PAGE TYPE
05ED 2076
05ED 2077
05ED 2078      INPUTS:
05ED 2079
05ED 2080      R0 = ACCESS MODE FOR PAGE OWNER CHECK
05ED 2081      R1 = PAGE TABLE ENTRY CONTENTS (NOT 0)
05ED 2082
05ED 2083      OUTPUTS:
05ED 2084
05ED 2085      R0 = STATUS CODE
05ED 2086          $$$ NORMAL IF SUCCESSFUL
05ED 2087          $$$ PAGOWNVIO IF CAN'T OPERATE ON THIS PAGE
05ED 2088      R1 = PAGE TYPE CODE IN LOW BYTE
05ED 2089          0 = TRANSITION
05ED 2090          1 = PAGE FILE
05ED 2091          2 = GLOBAL
05ED 2092          3 = SECTION INDEX
05ED 2093          4 = VALID
05ED 2094
05ED 2095
05ED 2096      *****
05ED 2097
05ED 2098      ***** THE FOLLOWING CODE MUST BE RESIDENT *****
05ED 2099
00000449 2100      .PSECT $MMGCODE
0449 2101
0449 2102      *****
0449 2103
0449 2104      MMG$PAGETYPE::
50 51 02 17 ED 0449 2105      CMPZV    #PTESV_OWN,#PTESV_OWN,R1,R0 ;PAGE OWNER OK?
      19 19 044E 2106      BLSS      20$ ;BRANCH IF NO
      50 01 3C 0450 2107      MOVZWL    #$$$ NORMAL,R0 ;SUCCESSFUL COMPLETION
      04 51 1F E1 0453 2108      BBC      #PTESV_VALID,R1,10$ ;BRANCH IF PAGE NOT VALID
      51 04 D0 0457 2109      MOVL      #4,R1 ;CODE FOR VALID PAGE
      05 045A 2110      RSB
51 51 05 16 EE 045B 2111 10$: EXTV    #PTESV_TYPO,#<PTESV_TYP1+1-PTESV_TYPO>,R1,R1
0460 2112
0460 2113      LOW BIT = TYPO, SIGN BIT = TYP1
0460 2114
      51 51 01 9C 0460 2115      ROTL     #1,R1,R1 ;BIT 0 = TYP1, BIT 1 = TYPO
      51 FC 8F 8A 0464 2116      BICB     #^C<3>,R1 ;CLEAR REST OF THE BYTE
      05 0468 2117      RSB ;AND RETURN THE PAGE TYPE
0469 2118
0469 2119      PAGE OWNER VIOLATION
0469 2120
      50 01EC 8F 3C 0469 2121 20$: MOVZWL    #$$$_PAGOWNVIO,R0 ;PAGE OWNER VIOLATION
      05 046E 2122      RSB
046F 2123
046F 2124
046F 2125
046F 2126      .END
```


SYSCREDEL
Symbol tableM 13
- SYSTEM SERVICE CREATE & DELETE PAGE16-SEP-1984 01:49:03 VAX/VMS Macro V04-00
5-SEP-1984 03:49:40 [SYS.SRC]SYSCREDEL.MAR;1Page 46
(22)

...PFN
ACCVIORET
ACMODE
BRBINC PGFLCNT
BRDELPAGRET
BRDELPAGSUCCESS
BUGS_CONTRACT
BUGS_IVGBLTYP
BUGS_NOSHMGSD
BUGS_WSLXVANMAT
CRECOM_DONE
CRECOM_INIT
CTLSGL_IPAGEFL
CTLSGL_PHD
CTLSGL_VIRTPEAK
DELAGAIN
DELAGAIN1
DELCOM
DELETFIRST
DELFALT
DELFALT1
DELGBLRDINPROG
DELGLOBAL
DELP1
DELPAGASTWAIT
DELPAGFIL
DELPAGIOWAIT
DELPAGIOWAIT2
DELPAGIOWAIT3
DELPAGIOWAIT4
DELPAGLAST
DELPAGLENVIO
DELPAGLOCKED
DELPAGNOTPROCES
DELPAGRESWAIT
DELPAGRET
DELPAGSUCCESS
DELPAG_WRTBAK
DELPAG_WRTDONE
DELPTE
DELPTE1
DELSTX
DELTRANS
DELVALID
EXESADJWSL
EXESALONONPAGED
EXESCNTREG
EXESCRTVA
EXESDELTVA
EXESEXPREG
EXESMAXACMODE
EXPANDCHK
FNDGBLSECTBL
INADR
INCPGFLCNT
IOCSGL_IRPFL
IPLS_ASTDEL

= 00000307 R 03
000002D5 R 02
= 0000000C
000002CA R 03
000000EC R 03
000000E9 R 03
***** X 02
***** X 03
***** X 03
***** X 03
000003A8 R 02
0000036B R 02
***** X 02
***** X 02
***** X 02
000002B0 R 03
00000289 R 03
00000322 R 02
00000402 R 02
000002AA R 03
00000354 R 03
0000012B R 03
000002CD R 03
00000148 R 03
00000204 R 03
000002BB R 03
0000022D R 03
00000241 R 03
00000262 R 03
00000252 R 03
000003F1 R 03
000000E4 R 03
000000C1 R 03
000000EF R 03
0000021C R 03
000003E9 R 03
000003E6 R 03
000003FC R 03
000001BC R 03
000003E0 R 03
0000008D R 03
000003C6 R 03
0000028D R 03
0000016D R 03
***** X 02
***** X 03
00000248 RG 02
00000187 RG 02
000002D9 RG 02
0000000E RG 02
***** X 02
000004D2 R 02
00000368 R 03
= 00000004
000003D8 R 03
***** X 03
= 00000002

IPLS SYNCH
IRPSB_RMOD
IRPSC_LENGTH
IRPSL_SEGVBN
IRPSW_SIZE
JIBSL_PGFLCNT
JIBSL_PGFLQUOTA
MFY_OR_FRELST
MMGSAB_DZRO
MMGS CONTRACT
MMGS CREDEL
MMGS CREPAG
MMGS CREPAGINI
MMGS CRETVA
MMGSC_LENGTH
MMGSDALCPAGFIL
MMGSDALCSTXSCN
MMGS DECSECREP
MMGSDELPAG
MMGSDELPFNLST
MMGSDELWSLEPPG
MMGS EXPREG
MMGS EXTRADYNWS
MMGS FAST_CREATE
MMGS FINDGSDPFN
MMGSGL_GBLSECFND
MMGSGL_GPTBASE
MMGSGL_MAXMEM
MMGSGL_SPTBASE
MMGSGL_SYSPHD
MMGSINADRINI
MMGSIN REGION
MMGSLCRLKPAG
MMGSL_CALLEDIPL
MMGSL_MAXACMODE
MMGSL_PAGESUBR
MMGSL_SAVRETADR
MMGSL_SVSTARTVA
MMGS MOVPTLOCK
MMGS MOVPTLOCK1
MMGSPAGETYPE
MMGSPGFLTWAIT
MMGSPTEREF
MMGSPURGWSSCN
MMGSREFCNTNEG
MMGSRELPFN
MMGSRESRCWAIT
MMGSRETADRINI
MMGSRETRANGE
MMGSSVPTX
MMGSSWAPWSLE
MMGSTRY ALL
MMGSV_CHGPAGFIL
MMGSV_DELGBLDON
MMGSV_NOWAIT IPLO
MMGSWRTPGSBAR
OPS_MOVL

= 00000008
= 0000000B
= 000000C4
= 00000048
= 00000008
= 0000003C
= 00000038
00000268 R 03
00000174 R 02
0000056D R 02
00000347 RG 02
00000414 RG 02
0000015C R 02
0000017F RG 02
= FFFFFFFE4
***** X 03
***** X 02
***** X 03
00000141 RG 03
***** X 03
***** X 03
00000000 RG 02
***** X 03
000000DD RG 02
***** X 03
***** X 03
***** X 02
***** X 03
00000297 RG 02
000001DF RG 02
***** X 03
= FFFFFFFF8
= FFFFFFFFC
= FFFFFFFF0
= FFFFFFFF4
= FFFFFFFEC
00000004 RG 03
00000000 RG 03
00000449 RG 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
000002A6 RG 02
000002C2 RG 02
***** X 03
***** X 03
0000008D RG 02
= 00000008
= 00000009
= 0000000A
***** X 03
= 000000D0

SYSCREDEL
Symbol table

- SYSTEM SERVICE CREATE & DELETE PAGE N 13

16-SEP-1984 01:49:03 VAX/VMS Macro V04-00
5-SEP-1984 03:49:40 [SYS.SRC]SYSCREDEL.MAR;1Page 47
(22)

OPS_MOVZWL	=	0000003C		
OPS_TSTL	=	000000D5		
OPS_TSTW	=	000000B5		
PAGCNT	=	00000004		
PCBSL_JIB	=	00000080		
PCBSL_PHD	=	0000006C		
PCBSW_DIOCNT	=	0000003E		
PCBSW_DIOLM	=	00000040		
PFNSAB_STATE	*****		X	03
PFNSAB_TYPE	*****		X	03
PFNSAL_BAK	*****		X	03
PFNSAW_REFCNT	*****		X	03
PFNSAX_SHRCNT	*****		X	03
PFNSAX_WSLX	*****		X	03
PFNSC_ACTIVE	=	00000007		
PFNSC_BADPAGLST	=	00000002		
PFNSC_FREPAGLST	=	00000000		
PFNSC_MFY PAGLST	=	00000001		
PFNSC_PPGTBL	=	00000004		
PFNSC_PROCESS	=	00000000		
PFNSC_RDERR	=	00000004		
PFNSC_RDINPROG	=	00000006		
PFNSC_RELPEND	=	00000003		
PFNSC_SYSTEM	=	00000001		
PFNSC_WRTINPROG	=	00000005		
PFNSM_DELCON	=	00000010		
PFNSM_MODIFY	=	00000080		
PFNSM_RPTEVT	=	00000040		
PFNSS_LOC	=	00000003		
PFNSS_PAGTYP	=	00000003		
PFNSV_LOC	=	00000000		
PFNSV_MODIFY	=	00000007		
PFNSV_PAGTYP	=	00000000		
PFNMAPDEL	=	0000006F	R	03
PHDSB_PAGFIL	=	0000001F		
PHDSL_FREPOVA	=	00000028		
PHDSL_FREP1VA	=	00000030		
PHDSL_FREPTCNT	=	0000002C		
PHDSL_MPINHIBIT	=	00000104		
PHDSL_POBR	=	000000C8		
PHDSL_POLRASTL	=	000000CC		
PHDSL_P1BR	=	000000D0		
PHDSL_P1LR	=	000000D4		
PHDSL_PSTBASOFF	=	00000020		
PHDSL_PTWSLELCK	=	00000064		
PHDSW_PSTLAST	=	00000024		
PHDSW_PTCNTLCK	=	0000003C		
PHDSW_PTCNTMAX	=	00000072		
PHDSW_WSDYN	=	0000000E		
PHDSW_WSLIST	=	00000008		
PHDSW_WSLOCK	=	0000000C		
PHDSW_WSQUOTA	=	00000018		
PHDSW_WSSIZE	=	00000050		
PRS_IPL	=	00000012		
PRS_POLR	=	00000009		
PRS_P1LR	=	0000000B		
PRS_TBIS	=	0000003A		

PSLSM_IPL	=	001F0000		
PTESC_EOWN	=	00800000		
PTESC_EW	=	28000000		
PTESC_KOWN	=	00000000		
PTESC_KW	=	10000000		
PTESC_SOWN	=	01000000		
PTESC_SW	=	40000000		
PTESC_UOWN	=	01800000		
PTESC_UW	=	20000000		
PTESM_MODIFY	=	04000000		
PTESM_PGFLVB	=	003FFFFFF		
PTESM_TYPO	=	00400000		
PTESM_TYP1	=	04000000		
PTESM_VALID	=	80000000		
PTESS_OWN	=	00000002		
PTESS_PFN	=	00000015		
PTESS_PGFLVB	=	00000016		
PTESV_PROT	=	00000004		
PTESV_CHKPNP	=	00000015		
PTESV_CRF	=	00000010		
PTESV_DZRO	=	00000011		
PTESV_MODIFY	=	0000001A		
PTESV_OWN	=	00000017		
PTESV_PFN	=	00000000		
PTESV_PGFLVB	=	00000000		
PTESV_PROT	=	0000001B		
PTESV_TYPO	=	00000016		
PTESV_TYP1	=	0000001A		
PTESV_VALID	=	0000001F		
PTESV_WINDOW	=	00000015		
REGION	=	00000010		
RETADR	=	00000008		
RSNS_ASTWAIT	=	00000001		
RSNS_NPDYMEM	=	00000003		
SCH\$GL_CURPCB	*****		X	02
SCH\$GQ_PFWQ	*****		X	03
SEC\$C_LENGTH	=	00000020		
SEC\$C_PAGCNT	=	0000001C		
SEC\$C_REFCNT	=	00000018		
SEC\$C_VXPFC	=	00000008		
SEC\$S_VPX	=	00000016		
SEC\$V_RESIDENT	=	0000000D		
SEC\$V_VPX	=	00000000		
SEC\$W_FLAGS	=	00000014		
SGN\$G_P1LWCNT	*****		X	02
SGN\$G_P1LWCNT	*****		X	02
SGN\$G_PTPAGCNT	*****		X	02
SHMPAGDEL	=	00000096	R	03
SS\$_ABORT	=	0000002C		
SS\$_ACCVIO	=	0000000C		
SS\$_EXQUOTA	=	0000001C		
SS\$_ILLPAGCNT	=	000000FC		
SS\$_INSFWSL	=	0000011C		
SS\$_NOPRIV	=	00000024		
SS\$_NORMAL	=	00000001		
SS\$_PAGOWNVIO	=	000001EC		
SS\$_VASFULL	=	00000244		

SYSCREDEL
Symbol table

- SYSTEM SERVICE CREATE & DELETE PAGE B 14

16-SEP-1984 01:49:03 VAX/VMS Macro V04-00
5-SEP-1984 03:49:40 [SYS.SRC]SYSCREDEL.MAR;1

Page 48
(22)

```
SS$_WASSET      = 00000009
TMP             = 00000001
UPDATPTCNT      = 00000549 R      02
VASM_P1         = 40000000
VASM_SYSTEM     = 80000000
VASM_VPG        = FFFFFFFE00
VASS_BYTE       = 00000009
VASS_VPN        = 00000015
VASV_P1         = 0000001E
VASV_VPN        = 00000009
WSL$M_PFNLOCK   = 00000010
WSL$M_WSLOCK    = 00000020
WSLXV$NOMAT     = 000001FC R      03
WSSCNWAIT       = 000000AB R      03
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
Y\$EXEPAGED	000005ED (1517.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$MMGCOD	0000046F (1135.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z\$INIT\$PFN_FIXUP_TABLE	00000018 (24.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:01.27
Command processing	127	00:00:00.52	00:00:06.91
Pass 1	476	00:00:18.55	00:01:02.77
Symbol table sort	0	00:00:02.78	00:00:07.93
Pass 2	364	00:00:05.74	00:00:18.28
Symbol table output	29	00:00:00.22	00:00:00.46
Psect synopsis output	2	00:00:00.01	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1035	00:00:27.90	00:01:37.66

The working set limit was 2100 pages.
115412 bytes (226 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1669 non-local and 125 local symbols.
2126 source lines were read in Pass 1, producing 38 object records in Pass 2.
38 pages of virtual memory were used to define 36 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_ \$255\$DUA28:[SYS.OBJ]LIB.MLB;1
_ \$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

24
9
33

1811 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYS CREDEL/OBJ=OBJ\$:SYS CREDEL MSRC\$:SYS CREDEL/UPDATE=(ENH\$:SYS CREDEL)+EXECML\$/LIB

0382 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSCANEVT
LIS

SYSCREPRC
LIS

SYSCHKPRO
LIS

SYSCREDEL
LIS

SYSCANCEL
LIS

SYSCOMMON
LIS

SYSCGMOD
LIS